LEARNING-BASED ADAPTIVE DESIGN FOR DYNAMIC SPECTRUM ACCESS IN COGNITIVE RADIO NETWORKS

By

Marjan Zandi

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND COMPUTER ENGINEERING
FACULTY OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY
OCTOBER 2014

Abstract

This thesis is concerned with dynamic spectrum access in cognitive radio networks. The main objective is designing online learning and access policies which maximize the total throughput of the secondary users in a cognitive radio network. As the first approach, we consider the auction-based formulation in design of dynamic spectrum access mechanisms where it is assumed that primary channels are heterogeneous with distinct availability statistics unknown to each secondary user (SU). Considering this approach, we first apply a unit demand (UD) auction which is called DGS (Demange-Gale-Sotomayor) auction. Applying the DGS auction, we explore the instantaneous link condition of each SU for its throughput maximization. To tackle the issues of this UD auction, we propose a learning-based unit demand (LBUD) auction. Our proposed auction mechanism incorporates a distributed learning of the primary channels into the auction mechanism to explore both primary channel availability statistics and instantaneous link gains of the SUs for their throughput maximization. This new mechanism substantially improves the communication overhead and also the SUs' throughputs where the primary channels have dissimilar availability statistics. The proposed LBUD auction preserves the strong property of the UD auction, *i.e.*, it is dominant strategy incentive compatible. To improve convergence speed of the iterative procedure used in the auction, we further propose an adaptive price increment algorithm. Simulation results show the effectiveness of the proposed LBUD auction mechanism in terms of the throughput gain.

As our second approach, we model the problem of designing decentralized dynamic spectrum access policies as a decentralized multi-armed bandit(DMAB) problem. Using DMAB formulation, we first propose a truly decentralized online learning and access policy where in addition to channel availability statistics, the secondary user population is also assumed to be unknown to the SUs. To reduce collision events at different learning stages, we then improve an existing access policy by exploiting a "perceived population" by each secondary user. We also develop a distributed learning and access policy which is effective in a wide range of primary channel conditions.

As our last approach, we investigate designing of a decentralized online learning and channel access in a cognitive radio network with M secondary users. We formulate the distributed channel selection problem in a cognitive network as a strategic game which is proved to be an exact potential game. Applying stochastic learning automata, we propose an adaptive decentralized access policy where each SU probabilistically selects one of the M-best channels to access. Based on collision events, we update the channel selection probability. In our proposed adaptive policy, two underlying distributed learning algorithms are utilized in parallel: i) Learning from sensing history on the primary channel availability, and ii) Learning from collision history on channel selections among SUs to avoid further collision. Simulation results show the effectiveness of our proposed adaptive policy in various distributions of mean channel availabilities across primary channels, as compared with other existing policies.

To my father whom I always miss

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof. Min Dong and Prof. Ali Grami for their continuous support, motivation, enthusiasm, and immense knowledge.

Oshawa, Ontario — Marjan Zandi
October, 2014

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

BLA	Bayesian Learning Automaton
CA	Capacity Assignment
CALA	Continuous Action-Set Learning Automata
CDF	Cumulative Distribution Function
CSMA	Carrier Sense Multiple Access
DLF	Distributed Learning with Fairness
DLP	Distributive Learning with Prioritization
DGS	Demange-Gale-Sotomayor
DMAB	Decentralized Multi-armed Bandit
DSA	Dynamic Spectrum Access
DSIC	Dominant Strategy Incentive Compatible
DSLPC	Discrete Learning Power Control
FALA	Finite Action-Set Learning Automata
i.i.d.	independent and identically distributed
KBLA	Kalman Bayesian Learning Automaton
LA	Learning Automata
LBUD	Learning Based Unit Demand
LHS	Left-Hand Side
MAB	Multi-armed Bandit
MIMO	Multi Input Multi Output

NE	Nash Equilibrium
ODE	Ordinary Differential Equation
OFDMA	Orthogonal Frequency Division Multiple Access
OSA	Opportunistic Spectrum Access
RHS	Right-Hand Side
RMAB	Restless Multi-armed Bandit
SARA	Stochastic Automata Rate Adaptation
SLA	Stochastic Learning Automata
SNR	Signal to Noise Ratio
SU	Secondary User
TABB	Two-Armed Bernoulli Bandit
TDM	Time-Division Multiplexing
UCB	Upper Confidence Bound
UD	Unit Demand
VCG	Vickery-Clarke-Groves

Chapter 1

Introduction

1.1 Static vs. Dynamic Spectrum Access

Radio spectrum is considered as one of the limited resources in spectrum management framework. In order to be allowed to operate in a specific frequency band, license is a necessity. Nowadays, radio spectrum usage is governed by a government agency in each country. Fixed spectrum allocation procedures were used as conventional techniques in which a certain frequency band is assigned to a user who has a license to operate in it. In recent years, due to the increasing demand for wireless spectrum, we have witnessed the inefficiency of the fixed spectrum allocation procedures. Under fixed spectrum allocation, the spectrum is not utilized efficiently since a large portion of the licensed spectrum is underutilized. To overcome the inefficiency of the spectrum utilization caused by static spectrum allocation, a more intelligent and flexible spectrum allocation paradigm, namely cognitive radio technology [1, 2], has been proposed.

1.1.1 Dynamic Spectrum Access in Cognitive Radio Network

A cognitive radio is an intelligent radio that senses the wireless spectrum for detecting available channels. It can adapt its operation by changing its transmission or reception operating parameters dynamically. In a hierarchical cognitive radio network, primary users which have the priority of using the spectrum, coexist with secondary users (SUs). The SUs need to perform spectrum sensing and exploit the unused spectrum whenever the primary users are inactive. Secondary users are not licensed to use the spectrum, therefore they can only opportunistically use the licensed spectrum when channels are idle. It is necessary that the SUs do not cause interference, above a predefined threshold, to the primary users [3].

To maximize the spectrum utilization in a cognitive radio network, an efficient dynamic spectrum sensing scheme is a requirement for recognizing the spectrum holes. Dynamic spectrum access (DSA) is the key concept in implementing the cognitive radio. DSA enables cognitive radio users (secondary users) to exploit the spectrum of the primary users [4–7].

We consider a cognitive radio network where M secondary users compete with each other to access one of the N available channels (see Fig. 1.1). Channel availability statistics are assumed to evolve as independent and identically distributed (i.i.d.) Bernoulli random processes with means unknown to the SUs. The main objective here is to design a distributed online learning and access policy which maximizes the total throughput of the secondary users.

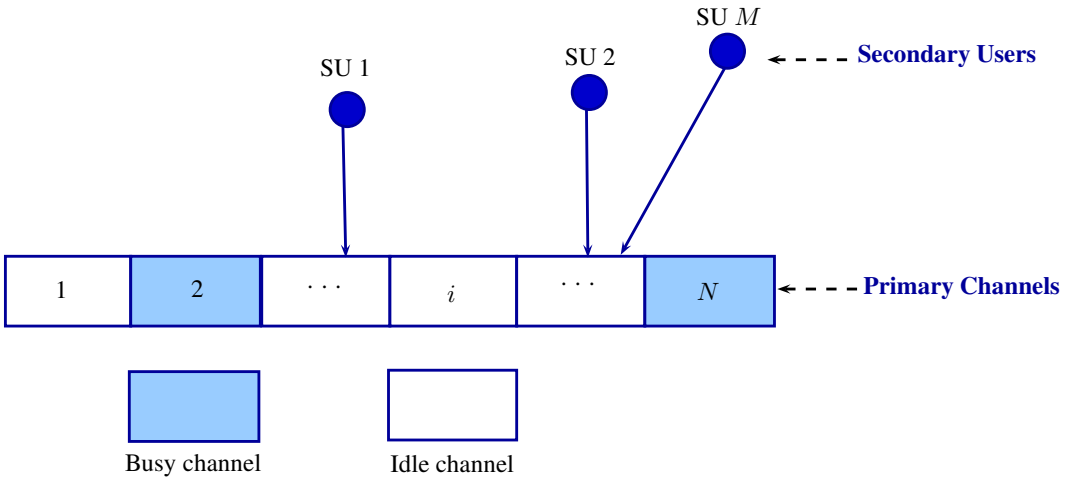

Figure 1.1: Cognitive radio network of N primary channels and M SUs.

1.2 Motivation

Inefficient usage of the spectrum is creating serious issues in wireless networks. We have observed the increase of demand for data rates and data services over the past few years. At the same time measurement studies have shown that many licensed spectrum bands have been remained unused. In order to utilize the licensed spectrum more efficiently and provide more effective communications, wireless networks need to be empowered by cognitive ability. In order to maximize its throughput, a cognitive radio network needs to have an efficient channel selection and access policy. In this thesis, our objective is to design access policies through developing decentralized online learning algorithms in a cognitive radio network. Extensive research has been conducted in this area using different approaches, *i.e.* auction, Multi-armed bandit (MAB), and Stochastic learning automata (SLA).

Auction-based approach compared with the contention-based distributed policies,

is a collision-free approach for spectrum access among SUs. Designing an auction for dynamic access will face several challenges: Each bidder (SU) needs to select which primary channels to bid for. The number of primary channels N might be large, and each channel has different loading statistics, therefore, bidding for all the primary channels may be undesirable in terms of both large communication overhead and poor throughput due to selecting an often busy channel. The question arises here is which channel or a subset of channels each SU should bid for, especially in case that channel availability statistics are unknown to the SUs. In addition, each SU's instantaneous link condition over the selected primary channel is directly reflected in its throughput. Existing auction-based distributed policies typically fail to consider such an important secondary link condition. In auction-based approach, to maximize the SUs' throughputs, both primary channel availability statistics and secondary link condition have been jointly taken into consideration in this thesis.

In the existing MAB-based approaches, although all distributed online learning and access policies can achieve logarithmic growth of regret (a measure of the difference between the total expected reward of the genie aided optimal decision and the expected reward obtained by a policy), but their relative performance is different due to different leading constants in the growth of regret. In our study we proposed adaptive policies which improved the leading constant of the normalized regret and provided substantial improvement over the existing policies. Furthermore, although there are some analysis of the existing policies on how the growth of regret changes with numbers of SUs and number of primary channels, no existing research has been conducted on how the distribution of mean availabilities of primary channels affects

the performance of these policies. In this thesis, we will show that among these existing policies, a policy might be more effective than the other policy for certain type of mean availability distribution of the primary channels, but not for other distributions. Designing learning and access policies that perform well for a wide range of primary channel mean availability distributions is practically desirable. In this thesis, we analyzed this effect and developed a learning and access policy that can be effective in various distributions of the mean channel availabilities.

In SLA-based approach on designing of distributed online learning and access policies in DSA context, we consider the primary network as the unknown random environment in which the primary channel availability statistics are unknown to the SUs. In this context, we refer SUs to the learning automata, the adaptive decision making agents. They intend to learn the optimum channel selection through a series of interactions with the random environment. SUs aim at determining the optimal actions out of a set of actions which are allowed. We proposed an adaptive policy which considers the two following types of underlying distributed learning: i). Learning from SU's own sensing history on the primary channel availability, and ii). Learning from SU's own collision history to adjust its channel selection among SUs for collision avoidance. Through jointly consideration of these two learning mechanisms, each SU's channel selection can adapt to different type of channel availability distributions across channels which is very desirable in practice.

1.3 Summary of Contribution

In this thesis, we devise dynamic spectrum access policies by applying the auction, MAB, and SLA formulations.

1. **Auction Formulation**

   Assuming heterogeneous primary channels with distinct availability statistics unknown to each secondary user, we consider the auction-based approaches to design dynamic spectrum access mechanisms. We first apply a unit demand (UD) auction proposed in [8] by exploring the instantaneous link condition of each SU for its throughput maximization. To address the disadvantages faced in the UD auction, we propose a learning-based unit demand (LBUD) auction. The LBUD auction incorporates a distributed learning of the primary channels into the auction mechanism to explore both primary channel availability statistics and instantaneous link gains of the SUs for their throughput maximization. The new mechanism not only substantially reduces communication overhead, but also improves the SUs' throughputs when the primary channels have dissimilar availability statistics. We show that the proposed LBUD auction preserves the strong property of the UD auction, *i.e.,* it is *dominant strategy incentive compatible*. We further propose an adaptive price increment algorithm to improve convergence speed of the iterative procedure used in the auction. Numerical results show the effectiveness of our proposed auction mechanism in terms of the throughput gain.

2. **MAB Formulation**

   It has previously been shown that problem of designing decentralized DSA policies can elegantly be modeled as a decentralized multi-armed bandit (DMAB) problem where M is known. Under MAB formulation:

   (a) We propose a truly decentralized online learning algorithm based on DMAB

problem for unknown user population M. We show that using distributed access policies with incorrect knowledge of M results in linear growth of regret, and underestimation incurs more significant loss than overestimation does. For distributed online learning of M, we propose a dynamic thresholding method, where the thresholds are dynamically determined using virtual systems built upon the current estimates of mean channel availabilities. Our algorithm allows both overestimation and underestimation in estimating M over time, and thus is capable of tracking the population change of secondary users.

(b) We consider the problem of decentralized online learning and channel access in a cognitive radio network. Based on an existing distributed access policy proposed in [9], named the ρ^{RAND} policy, we propose an adaptive decentralized access policy in which the distributed coordination among secondary users is adjusted at different stages of learning accuracy of the primary network. Specifically, we exploit a "perceived population" by each secondary user to reduce collision events at different learning stages. We design a metric that measures the level of learning accuracy and use that as an indicator to adjust the "perceived population" by each secondary user. Simulations show that our proposed adaptive policy improves the leading constant of the normalized regret and can provide substantial improvement over the ρ^{RAND} policy.

(c) We consider the effect of the mean availability distribution of primary channels on the performance of distributed learning and access policies, and develop a distributed learning and access policy that is effective in a

wide range of primary channel conditions. We first extend the recently proposed Bayesian learning automata (BLA) algorithm to distributed online learning of underlying primary channel availabilities, and modify the existing access policies to form BLA-ρ^{RAND} and BLA-DLF policies. By analyzing the distributed access collision mechanism offered by the ρ^{RAND} and DLF policies [9, 10], we identify how different mean channel availability distributions can impact the effectiveness of each policy. In light of this, we propose DSLA policy that adapts to different channel availability distribution conditions. Based on a closeness factor we propose, the DSLA policy automatically switches between the underlying learning policies, as well as the access policies, to determine which policy is the most effective for a given primary channel condition. Simulation studies show that our proposed DSLA policy is effective in providing a good performance for a wide range of primary channel availability distributions.

3. SLA Formulation

We focus on designing a decentralized online learning and channel access in a cognitive radio network with M secondary users. The distributed channel selection problem in this network is formulated as a strategic game which is proved to be an exact potential game. In this thesis, we aim at designing an adaptive policy that can effectively respond to different primary network conditions. We propose an adaptive decentralized access policy by applying stochastic learning automata where each SU probabilistically chooses one of the M-best channels to access. The channel selection probability is then updated based on collision

events. The following two underlying distributed learning algorithms are utilized in our proposed adaptive policy: 1. Learning from sensing history on the primary channel availability, and 2. Learning from collision history on channel selections among SUs to avoid further collision. Some previously proposed distributed access policies can be viewed as special cases of our proposed adaptive policy, with a set of pre-set channel selection probabilities. Numerical results demonstrate the effectiveness of our proposed adaptive policy in various distributions of mean channel availabilities across primary channels, as compared with other existing policies.

1.4 Organization of the Thesis

The rest of the thesis includes five chapters. In Chapter 2, we introduce dynamic spectrum access using auction, multi-armed bandit, and stochastic learning automata approaches. In Chapter 3, we consider the auction-based approaches for dynamic spectrum access with unknown primary channel availability statistics to the SUs. we propose the LBUD auction, in which distributed learning of the primary at each SU is performed and incorporated into the auction mechanism. Chapter 4 considers dynamic spectrum access using multi-armed bandit framework. In Chapter 5, we investigate the problem of decentralized online learning and channel access in a cognitive radio network through a game theoretic approach using stochastic learning automata. In Chapter 6, we discuss some future research directions and conclude the thesis.

1.5 Notations

We represent the statistical expectation by $\mathrm{E}[\cdot]$. We use lowercase and uppercase boldface letters to represent the vectors and matrices, respectively. Also, $|\cdot|$ stands for cardinality of a set and $\lceil\cdot\rceil$ represents a ceiling function.

Chapter 2

Background and Literature Review

2.1 Dynamic Spectrum Access via Auction Approach

Transition of wireless networks from static, centralized, and homogeneous networks to dynamic, distributed, and heterogeneous networks adds new complex challenges to designing of the spectrum access policies. Spectrum access in new dynamic wireless environment cannot be supported by traditional fixed spectrum allocation schemes. To overcome these new challenges, economics approaches, such as auction, have also been employed to spectrum access mechanisms. An auction [11] is considered as a mechanism for allocating resources among several bidders. Auction mechanisms are built based on the concept of selling and buying the objects which are auctioned. In DSA context, auction mechanisms are used as a promising approach for fully utilizing the spectrum.

There are three main parts involve in an auction model:

1. Auctioneer (coordinator): Typically, an auctioneer is required in an auction.

The auctioneer which coordinates the auction is either integrated in the seller or is an independent trusted third party.

2. Bidder: A bidder of an auction is the one who is willing to purchase the objects in an auction.

3. Item (commodity): An auction item is an object that is auctioned to be sold.

In an auction, valuation of a bidder on an object is referred to how that specific object worths to the bidder. Based on their preferences, bidders might value objects of an auction differently. Auctions are specifically used because the seller has no idea about the bidders' valuations (the maximum amount each bidder is willing to pay) on the objects. If these valuations are known to all bidders in an auction, it is a public valuation. Private valuations are referred to those valuations that are unknown to the other bidders. In an auction process, the auctioneer indicates an asking price on objects to be sold. The asking price on an object is not generally accepted by a bidder whether the asking price exceeds its valuation on that specific object.

Traditional auction theory has concerned with sealed-bid for a single object. Unlike open-cry auctions, in sealed-bid auctions, without knowledge of any of their opponents' bids, bidders of the auction submit their sealed bids to the auctioneer. After receiving the sealed bids, based on rule of the auction, the auctioneer determines the winner and thus the single object is assigned to the winner. First-price auction and second-price auction [12] are the two most widely investigated sealed-bid auctions for a single object. In first-price auction, the bidders submit their bids simultaneously to the auctioneer. The auctioneer then assigns the single object to the bidder with the highest bid. The winner needs to pay the amount equal to its bid. In second-price

auction, the bidders also submit their bids simultaneously to the auctioneer. The auctioneer then assigns the single object to the bidder with the highest bid. Unlike the first-price auction, the winner needs to pay the amount equal to the second highest bid.

Different from MAB formulations, auction design has recently attracted interests for dynamic spectrum access in cognitive radio networks [13–19]. Treating available channels as objects and SUs as bidders, the channel selection or assignment can be made through an auctioning process. Compared with the contention-based distributed policies described earlier, it is a collision-free approach for spectrum access among SUs. Several challenges are faced in designing an auction for dynamic access: Each SU needs to decide which primary channels to bid for. Since the number of primary channels N can be large, and each channel has different loading statistics, bidding for all the primary channels may be undesirable in terms of both large communication overhead and poor throughput due to selecting an often occupied channel. Then, the question is which channel or a subset of channels each SU should bid for, especially when channel availability statistics are unknown to the SUs. Moreover, each SU's throughput directly links to its own instantaneous link condition over the selected primary channel. Although important, such a secondary link condition is typically ignored in the existing distributed policies mentioned above. However, to maximize the SUs' throughputs, both primary channel load and secondary link condition should be taken into account in the auction process for channel selections. In addition, each SU intends to maximize its own payoff through bidding. Thus, it is desirable for the auction mechanism to possess some nice properties for individual performance guarantee.

2.1.1 Related Works

As mentioned earlier, using a decentralized MAB formulation, a few decentralized learning and access policies were proposed [9,10,20–23]. These existing access policies make channel selection and access solely based on the estimated mean availabilities of primary channels from sensing history, but do not explore the instantaneous fade conditions in the secondary user transmission links.

Game-theoretic approaches have been considered for designing channel selection and access policies in cognitive radio networks [24–28], where SUs' accesses have been modeled and formulated using certain type of games. A joint consideration of primary channel availability statistics and instantaneous link gains of the SUs is not considered in the model and game formulation of these works.

Auction-based approaches have recently attracted many research interests for efficient spectrum access, sharing, or leasing [13–19, 29]. Many of these works treat primary channels as multiple objects for SUs to bid, and allow each SU to be assigned multiple channels to maximize certain defined utility [13, 16–19], instead of requesting only one channel (unit demand). In [16], spectrum trading in TV band is considered by taking into effect of imperfect spectrum sensing. To handle bidding for multiple objects, a multi-unit sequential sealed-bid first-price auction is proposed. Spectrum auction with multiple primary spectrum auctioneers is considered in [17], and a progressive auction is proposed for each SU to select its best spectrum auctioneer for bidding. In [18] an auction-based cooperative sensing protocol for SUs is proposed. In [19], VCG-based auction mechanisms were proposed for joint interference control and spectrum auction for SUs with mobility. In [13], based on the

second-price auction [12], a repeated auction is considered to determine the assignment of available channels to SUs based on certain cost utility function. Bertsekas auction algorithm [30] was proposed as a fast-converging algorithm for the assignment problem. It can be applied to the channel assignment among SUs. To reduce the high communication overhead incurred in the Bertsekas auction algorithm, [29] modified the Bertsekas algorithm and proposed a fully distributed auctioneer-free auction algorithm by using an opportunistic carrier sense multiple access (CSMA) assignment scheme. Besides the above mentioned single-side auctions focusing on bidding among SUs, treating both secondary and primary users as bidders for channel access, double-sided auction is considered in [14, 15].

In all the above works, either the primary channels for auction are assumed available or they are assumed homogeneous in nature without taking into account the different loading conditions (*i.e.,* availability statistics) and their impact on the channel assignments. In addition, different from our problem, each SU is allowed to win multiple channels depending on the auction outcome, instead of each SU selecting one channel to access. Furthermore, the second-price (Vickery) auction [12] or the Vickery-Clarke-Groves (VCG) auction [31–33] adopted in the existing works are designed for bidding a single object and multiple objects, respectively.

For auction design in economics, the second-price auction was proposed for bidding a single object[1], and was shown to be DSIC. It also achieves the minimum price equilibrium. This auction has then been generalized to the VCG auction for the scenario with multiple objects. For bidding multiple heterogeneous objects with unit demand, the DGS auction was proposed [8], which was shown to be DSIC. Like

[1]In second-price auction, the bidder with the highest bid is winner of the auction and the amount it has to pay is equal to the second highest bid.

the VCG auction, the DGS auction also preserves some interesting properties of the second-price auction, *i.e.,* it is DSIC and achieves the minimum price equilibrium. In addition to these common properties, the DGS auction has extra nice properties which motivated us for considering this auction in our work. In practice, bidders may have limited budgets for auction. It is shown in [34] that addressing budgets properly breaks down the incentive compatibility of the VCG auction, while in [35] it is shown that the DGS auction is incentive compatible even if the bidders have budget constraints. In addition, the DGS auction mechanism is group strategy-proof, while the VCG auction is vulnerable to collusion [36, 37]. Due to these properties for the DGS auction, in our work, we consider the DGS auction for bidding the primary channels with heterogeneous channel availability statistics.

In the traditional auction theory, it is also assumed that all bidders of the auction are capable of paying up to their valuations on the objects of the auction. In practice, there are some bidders that might have budget constraints [35,38]. The authors of [38] considered the case where heterogeneous items are auctioned to multiple bidders with limited budgets. To approach this problem, they generalized the DGS auction from setting without financial constraints to settings with financial constraints. In [35], it is shown that the DGS auction is incentive compatible even though with budget constraint bidders.

2.2 Dynamic Spectrum Access via Multi-armed Bandit Approach

Classical MAB [39–41] is a bandit consists of a single player and N independent arms. The arm $i \in \{1, ..., N\}$ which is played receives an i.i.d. random reward with an unknown mean θ_i. At each time, the player selects one arm to play. The goal is to maximize the total expected reward in the long run. Multi-armed bandit problem is considered as a problem in reinforcement learning[2], an area of machine learning, which models the trade-off between exploitation (immediate reward maximization) and exploration (gaining new information). The objective of a bandit problem lies in determining the optimal balance between exploitation and exploration. Modelling the trade-off between exploitation and exploration by the bandit problems is fundamental in a variety areas of research including DSA. In the DSA context, the problem is how the SUs choose between several different primary channels. In this context, the SUs are considered as the players and the primary channels are considered as the arms.

We consider a cognitive radio network with N independent channels and M secondary users, where $N \geq M$. In the DSA context, the classical MAB can be used to formulate the problem of selecting the M-best channels under unknown channel availability statistics in centralized scheduling of users' access such that through this selection the total throughput of the secondary users is maximized. In this case, the problem is to design a policy to sequentially choose M plays of N arms with i.i.d.

[2]Reinforcement learning is an area of machine learning which is concerned with agents performing actions in an environment to maximize reward. The environment provides feedback (reinforcement) to the agents regarding their actions. The agents then adjust their actions based on the received feedback.

rewards over time. For distributed access by the secondary users, the problem formulation can be viewed as the decentralized MAB (DMAB) problem. In contrast to the classic MAB problem, for decentralized MAB problem, M players compete over N arms. When multiple secondary users select the same arm, collision occurs resulting in lost rewards. To address this problem, which particularly arises in dynamic spectrum access, several decentralized learning and access policies have been recently developed [9, 10, 20]. These policies use different mechanisms to achieve "coordination" among secondary users to orthogonalize their access to the M-best channels, and all achieve logarithmic growth of regret. Performance of a MAB policy is evaluated by a common measure called regret. The regret of a MAB algorithm is defined as the difference between the total expected reward of a genie aided algorithm (which always makes the optimal decision) and the expected reward obtained by the algorithm.

Common to all these proposed distributed algorithms is the assumption that the number of secondary users (M) is known to each secondary user. This information is utilized in determining the access decision to one of the M-best channels. Thus, although these algorithms are distributed in terms of learning the channel availability statistics based on local observation histories, secondary users share the common knowledge of the user population. In a practical dynamic environment, such knowledge may not be known and needs to be estimated and tracked at each secondary user in order to implement the distributed access policy.

As it was mentioned, each channel is associated with an i.i.d. stochastic reward with an unknown mean to the secondary users. Secondary users need to learn the unknown means from their local observations. The secondary users may not exchange information. The goal is to develop a sequential online learning policy, running at

each secondary user, which enables that secondary user to make a sensing and access decision among all N channels such that through this serial of decisions, the total expected reward is maximized.

2.2.1 Stationary and Dynamic MAB

Multi-armed bandit problems can be categorized as *stationary* (non-Bayesian) or *dynamic* (Bayesian or non-stationary) bandit problems. In stationary bandit problems, rewards offered by different arms are fixed but unknown parameters $\boldsymbol{\theta} \triangleq [\theta_1, \cdots, \theta_N]^T$ which need to be learned through online learning processes. In stationary bandit problems, it is also assumed that the number of arms and players are fixed through the whole learning process. In dynamic bandit problems, however, rewards are not static and depend on other parameters such as time and/or a time varying set of arms or players. In dynamic MAB, due to involvement in a changing environment, new challenges are added to the bandit problems. Extensive research has been conducted in both areas of stationary and dynamic bandit problems.

2.2.2 Exploitation vs. Exploration

Through online learning processes at each time slot, each SU confronts the trade-off between exploitation and exploration. Exploitation considers selecting the channel with the maximum expected reward based on its local observation (gaining the best immediate reward) while exploration considers selecting other channels in order to gain more information and learn about their unknown parameters. The objective of the explorations lies in the learning the unknown reward model for all arms to minimize the risk of selecting an inferior arm in the future. Considering this trade-off will

raise the following question: How the secondary users learn the unknown parameters and make their own decisions while there is a competition among cognitive users? In other words, when do the cognitive secondary users decide to take different actions rather than gaining their best immediate rewards, *i.e.*, when is the best time for secondary users to do the exploration instead of exploitation? Exploration, spending time to learn the unknown parameters of the different options rather than selecting the best immediate reward, is costly. The beauty of the MAB problem lies in the fact that it can elegantly find the balance between exploration and exploitation.

2.2.3 Learning the Unknown Means

To develop an efficient sequential online learning policy, numerous studies have been conducted. Among the proposed policies, there are policies in which the learning of unknown means depends on Upper Confidence Bound 1 (UCB1) and BLA algorithms.

UCB1

In the UCB1 algorithm, a sample-mean based index policy for the single user case, an index is assigned to every channel. The assigned index is a statistic which is based on the estimated sample mean of channel i and the total number of times that channel i has been visited up to the current time slot n. Let $T_i(n)$ denote the number of times that the secondary user senses channel i up to time slot n. If the secondary user selects channel i to sense at time slot n, then it obtains the value of $X_i(n)$ and records this value as $X_i(T_i(n))$.

$$X_i(n) = \begin{cases} 1, & \text{if channel } i \text{ is available at time slot } n \\ 0, & \text{otherwise} \end{cases} \tag{2.2.1}$$

Let $\mathbf{X}_i(n) \triangleq [X_i(1), \cdots, X_i(T_i(n))]^T$ be the vector holding the sensing observations of the secondary user for channel i up to time slot n. With these sensing observations, the secondary user can estimate θ_i, the mean availability of channel i, at time slot n as

$$\hat{\theta}_i(T_i(n)) \triangleq \frac{1}{T_i(n)} \sum_{k=1}^{T_i(n)} X_i(k). \tag{2.2.2}$$

The secondary user obtains an index called g-statistic for all the i^{th} channels, $i = 1, \cdots, N$, as

$$I_i(n) \triangleq \hat{\theta}_i(T_i(n)) + \sqrt{\frac{2 \log n}{T_i(n)}} \tag{2.2.3}$$

In the UCB1 algorithm, the above index will be used to rank all the channels and the user then selects the channel with the highest index at time slot n.

BLA

The BLA algorithm proposed in [42] considers a classic stationary two-armed Bernoulli bandit (TABB) problem. The BLA algorithm is an efficient algorithm constructed based on the Bayesian inference (see [43] for detail). This algorithm uses the conjugate prior distributions and it has highly computationally efficient updating rules. Updating rules rely on updating the hyper parameters associated with the conjugate prior distributions. In [42], it is shown that this algorithm achieves logarithmic scaling of the regret and outperforms upper confidence bound tuned (UCB-Tuned) algorithm [44] for all reward distributions except the case that rewards of the two arms

are coming from distributions with high variance combined with the small difference between their expected rewards.

In the BLA algorithm, a beta distribution with two positive parameters α and β is assigned to each single arm. Probability density function of the corresponding beta distribution is denoted as

$$f_i(z;\alpha_i,\beta_i) \triangleq \frac{z^{\alpha_i-1}(1-z)^{\beta_i-1}}{\int_0^1 y^{\alpha_i-1}(1-y)^{\beta_i-1)}dy}, z \in [0,1], i \in [1,2] \tag{2.2.4}$$

Below, we summarize the BLA algorithm:

1. Two beta distributions f_1 and f_2 are produced to be assigned to the two arms.
2. Two sets of the positive parameters (α_1,β_1) and (α_2,β_2) are initialized to $(1,1)$.
3. Two random variables z_1 and z_2 are randomly drawn from two produced beta distributions f_1 and f_2.
4. Two random variables z_1 and z_2 are compared, and corresponding arm s with the largest random variables is selected.
5. If we receive reward from selecting arm s, then α_s is increased by one otherwise β_s is increased by one.

It is noteworthy that the BLA approach can be extended to a multi-armed bandit problem and it can be applied to a dynamic multi-armed bandit problem as well [42].

2.2.4 Related Work

MAB problem was first proposed in [45] for its application in the context of clinical trial. Under the Bayesian formulation, [46,47] showed that an index policy is optimal

for the classic MAB. In the proposed index policy, a priority index which is called the Gittins index is assigned to each state of the arms. Then, the arm whose current state has the largest index is activated as the result of the optimal action. It is also shown that the proposed index policy reduces the complexity of the MAB problem from exponential to linear with the number of arms.

Under Bayesian framework, classic MAB has been extended to the Restless Multi-armed Bandit (RMAB) [48]. In RMAB, multiple arms are allowed to be activated simultaneously and passive arms are also allowed to change states and offer rewards. Generalization of the classic MAB to RMAB significantly expands application area of the MAB [49]. Implementing Whittle index policy is difficult due to complexity of establishing its existence (indexability), computing the index, and establishing its optimality in the finite regime.

An extensive research has also been conducted in the area of the RMAB. RMAB has been studied in numerous studies [49–56].

In [54], a restless multi-armed bandit is investigated, where channels are modeled as independent and identical GilbertElliot channels with imperfect channel state detection. In [55], the distributed learning problem in cognitive radio networks is formulated as DMAB where the channel state observation is imperfect.

One of the earliest results on classic stationary MAB problem has been presented by Lai and Robbins [39]. In this paper, the authors consider a classical MAB model with a single player, a single play, and N arms each offering independent and identically distributed stochastic reward with means unknown to the player. A sequential online learning policy is then designed with the objective to maximize the overall expected reward. This policy achieves logarithmic scaling of the regret over time, but

is linear in the number of arms.

Anantharam *et al.* [40] developed a single player MAB policy with multiple plays. This policy is an extension of the Lai and Robbins policy and can be considered as a centralized MAB with multiple players who can jointly make decisions based on their joint observations. In [40], in contrast to the classic MAB, multiple arms can be played at the same time. It is shown that this policy also achieves logarithmic scaling of the regret by a larger leading constant as compared to that of Lai and Robbins policy.

In [42], Granmo considered a classic stationary two-armed Bernoulli bandit (TABB) problem and proposed an efficient algorithm called the BLA algorithm which is constructed based on the Bayesian estimation. The BLA algorithm uses the conjugate prior distributions with highly computationally efficient updating rules. Updating rules rely on updating the hyper parameters associated with the conjugate prior distributions. [42] shows that this algorithm achieves logarithmic scaling of the regret. It is also shown that the BLA algorithm outperforms upper confidence bound tuned (UCB-Tuned) algorithm [44] for all reward distributions except the case that rewards of the two arms are coming from distributions with high variance combined with small difference between their expected rewards. This approach can be extended to a multi-armed bandit problem and it can be applied to a dynamic multi-armed bandit problem as well [42].

Later, Braadland and Norheim [57] extended the Granmo's work in [42]. They introduced three new algorithms to the BLA family, called BLA Poisson, BLA Normal known δ^2, and BLA Normal unknown δ^2. They also empirically evaluated the set of BLA family, and compared their performances with their competing schemes in

MAB problem. The three proposed algorithms are designed for Bernoulli, Poisson, and normally distributed rewards. In this work, [57] shows that the BLA family outperforms all other learning schemes in multi-armed bandit problem.

Stian Berg [43] also extended Granmo's work [42] by applying Bayesian estimation and introduced a new approach for a dynamic multi-armed bandit problem. Applying Kalman filtering on dynamic multi-armed bandit problem, the author of [43] proposed an algorithm called Kalman Bayesian Learning Automaton (KBLA) which outperforms the previously introduced solutions in a non-stationary environment.

Recently, an extensive research has been conducted in the area of the DMAB. DMAB in the context of the opportunistic access in a cognitive radio network has been studied in numerous studies [9, 10, 20, 58–60].

Granmo *et al.* [61] proposed a stationary decentralized two-armed bandit algorithm based on the goore game while each player is inherently Bayesian in nature.

Anandkumar *et al.* [59] proposed a decentralized MAB algorithm called ρ^{RAND} policy. ρ^{RAND}policy cannot be considered as a truly distributed policy since the total number of secondary users is assumed to be known to each cognitive user. This algorithm is based on the UCB1 algorithm proposed by Auer *et al.* [44] which is an index based policy. The proposed policy in [59], ρ^{RAND} policy, extends the single player UCB1 policy to a policy with multiple players which do not exchange information among themselves.

After the policy proposed by Lai and Robbins [39], several simpler index based policies [44, 62] were developed. In these policies, at every time slot an index is assigned to every single arm. The assigned index is a statistic which is based on the estimated sample mean and also on the total number of times that a particular arm

has been played up to the current time slot. The single player obtains an index called g-statistic for all the arms and then selects an arm with the highest amount of the g-statistic.

Anandkumar *et al.* [9] proposed another decentralized policy which relaxes the previous assumption indicating that the total number of secondary users should be known to each cognitive user. At each time slot, in their proposed policy, called ρ^{EST} policy (which is an extension of the ρ^{RAND} policy), the total number of secondary users needs to be estimated first and then based on the updated version of the total number of secondary users, ρ^{RAND} policy needs to be implemented. Implementing ρ^{EST} policy, each secondary user updates total number of the secondary users based on the total number of collisions experienced by itself up to the current time slot. In this work, they consider a threshold which is a function of estimated number of secondary users and the time horizon. They use this threshold against the total number of collisions experienced by the secondary user through process of updating the estimated number of secondary users. More specifically, every time that the total number of collisions occurred to each secondary user exceeds the obtained threshold, estimated number of thresholds is incremented by one.

Most of the previous work on DMAB have relied on assumption that each secondary user sees the same channel availability statistics. In contrast with this assumption, Gai *et. al* [60] considered the case where cognitive users see different channel availability statistics. The authors of [60] showed that their problem formulation results in dependent arms. Gai *et al.* [10] recently proposed two different DMAB algorithms. One of these algorithms considered the prioritized access problem while the other algorithm considered the fair access problem in MAB. Distributive learning

with prioritization (DLP) algorithm assigns a prioritization rank to each cognitive user. The goal of this algorithm is that the cognitive user with rank k selects the channel with k^{th} highest expected reward. Since both the total number of cognitive users and also the prioritization rank is assumed to be known, this algorithm cannot be considered as a truly decentralized algorithm. Their second proposed algorithm, the distributed learning with fairness (DLF) algorithm, takes care of the fairness among all the cognitive users meaning that applying this policy, each cognitive user can receive the same expected reward. Both DLP and DLF are constructed based on the UCB1 policy.

2.3 Dynamic Spectrum Access via Stochastic Learning Automata Approach

2.3.1 Single LA

A learning automaton is an adaptive decision making device deployed in an unknown environment. The LA needs to learn the unknown environment through its repeated interactions with it. This interactions are achieved through selecting an action from a set of actions by the LA. Therefore, the LA obviously needs to select the optimum action out of a set of possible actions. The learning automaton chooses the optimum action according to a specific probability distribution which is updated based on the response of the random environment after performing a specific action. Based on the set of actions which an automaton can take an action from, the LA is classified to the following two types:

1. **FALA**: Finite action-set learning automata (FALA) are referred to the learning automata where the size of their corresponding action-set is finite.

2. **CALA**: Continuous action-set learning automatoa (CALA)are referred to the learning automata deal with a continuous action space.

An LA also can be classified to three different models based on the input obtains from the unknown environment as follows:

1. **P-model**: If an LA has an binary input set, e.g., $\{0, 1\}$, it is a P-model LA.

2. **Q-model**: If an LA has an input set with a finite collection of distinct symbols, it is a Q-model LA.

3. **S-model**: If an LA has an input set from an interval $[0, 1]$, it is a S-model LA.

Learning Process

Let $\mathcal{C} = \{1, ..., N\}$ denote the set of possible actions and $\alpha(n) \in \{1, ..., N\}$ is the action that is randomly taken by the LA at time slot n. The LA takes an action based on its action probability distribution $\mathbf{p}(n) = [p_1(n), ..., p_N(n)]$ where $p_i(n)$ is the probability that action $i = \alpha(n) \in \{1, ..., N\}$ is taken at time slot n by the LA ($p_i(n) = \text{Prob}[\alpha(n) = i]$ and $\sum_{s=1}^{N} p_s(n) = 1, \forall n$). The selected action by the LA is fed to the environment. The response of the environment, a stochastic reaction or feedback (reinforcement), to the selected action is returned to the LA as an input. Let $\Upsilon(n)$ denote the corresponding reinforcement. In this thesis, we consider P-model FALA, therefore, $\Upsilon(n) \in \{0, 1\}$. Now let ϱ_i denote the expected value of $\Upsilon(n)$ given that $\alpha(n) = i$ ($\varrho_i = \text{E}[\Upsilon(n)|\alpha(n) = i]$). ϱ_i is referred to the reward probability

corresponding to action $\alpha(n) = i, i \in \mathcal{C}$. Let $\boldsymbol{\varrho} = [\varrho_1, ..., \varrho_N]$ denote the reward probability vector.

The reward probabilities are unknown to the LA. Otherwise the LA would obviously select the action with the highest reward probability as its optimal action. The problem here is how an LA can select the optimal action while it has no knowledge on the reward probabilities to maximize the expected value of the reward obtained from the environment. Using a learning algorithm, this problem can be easily solved. A single LA is shown in Fig. 2.1.

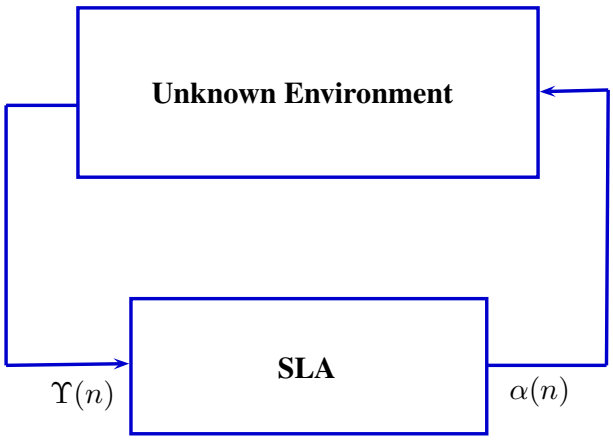

Figure 2.1: Single LA.

Updating Scheme

Through the interactions with the unknown environment, a learning algorithm (updating scheme or reinforcement scheme) such as Linear Reward-Inaction scheme [63, 64], initially was considered in mathematical psychology, is used to update the action probability distribution $\mathbf{p}(n)$. This scheme then has been introduced to the engineering field [65, 66].

The concept behind this scheme is intuitively easy to understand. Assuming upon selecting action i ($i = \alpha(n)$) as the optimum action by the LA at time slot n, it receives a reward ($\Upsilon(n) = 1$). The corresponding action probability $p_i(n)$ is increased and the rest ($\forall\ p_s(n), s \in \mathcal{C}$ and $s \neq i$) are decreased. Under this scheme, the action probabilities will not be affected where a penalty ($\Upsilon(n) = 0$) is received by the LA. The updating rule in Linear Reward-Inaction scheme is defined as

$$\begin{aligned} p_i(n+1) &= p_i(n) + \Upsilon(n)(1 - p_i(n)), \quad \text{for } i = \alpha(n) \\ p_i(n+1) &= p_i(n) - \Upsilon(n)p_i(n), \quad \text{for } i \neq \alpha(n) \end{aligned} \tag{2.3.1}$$

Since in this scheme the updated action probabilities $\mathbf{p}(n+1)$ is a linear function of the components of the $\mathbf{p}(n)$, it is called a linear scheme. The nonlinear and hybrid schemes are investigated in [67, 68]. Based on the reward probabilities, the unknown environment can be either stationary or non-stationary. An unknown environment is called a stationary environment where the reward probabilities do not depend on n and an unknown environment is called a non-stationary environment where the reward probabilities depend on n. In this thesis, we consider a stationary environment.

2.3.2 Multiple LA

Up until now, we have discussed a single learning automaton which interacts with a random environment to learn the unknown reward probabilities. In practice, we deal with multiple learning automata that try to learn an unknown environment either cooperatively or competitively. The operating of several learning automata on an environment will result in a game of learning automata [69]. In this game, the action probabilities belonging to each learning automaton are independently updated based on the selected learning scheme.

Learning Process

Let M denote number of learning automata performing in the same environment. Let also $\mathcal{C} = \{1, ..., N\}$ denote the set of possible actions and $\alpha^j(n) \in \{1, ..., N\}$ is the action that is randomly taken by the LA j at time slot n based on its action probability distribution $\mathbf{p}^j(n) = [p_1^j(n), ..., p_N^j(n)]$ where $p_i^j(n)$ is the probability that action $i = \alpha^j(n) \in \{1, ..., N\}$ is taken at time slot n by the LA j ($p_i^j(n) = \text{Prob}[\alpha^j(n) = i]$ and $\sum_{s=1}^{N} p_s^j(n) = 1, \forall n$). The selected actions by the learning automata are fed to the environment. The responses of the environment, stochastic reactions or feedbacks (reinforcements), to the selected actions are returned to the learning automata as their inputs. Let $\Upsilon^j(n), \in \{0, 1\}$ denote the corresponding reinforcement. Now let ϱ_i^j denote the expected value of $r^j(n)$ given that $\alpha^j(n) = i$ ($\varrho_i^j = \text{E}[\Upsilon^j(n)|\alpha^j(n) = i]$). ϱ_i^j is referred to the reward probability of LA j corresponding to action $\alpha^j(n) = i, i \in \mathcal{C}$. Let $\boldsymbol{\varrho}^j = [\varrho_1^j, ..., \varrho_N^j]$ denote the reward probability of LA j. A multiple LA is shown in Fig. 2.1.

Updating Scheme

The updating rule under Linear Reward-Inaction scheme is defined as

$$\begin{aligned} p_i^j(n+1) &= p_i^j(n) + b\Upsilon^j(n)(1 - p_i^j(n)), \quad \text{for } i = \alpha^j(n) \\ p_i^j(n+1) &= p_i^j(n) - b\Upsilon^j(n)p_i^j(n), \quad \text{for } i \neq \alpha^j(n) \end{aligned} \tag{2.3.2}$$

where $\Upsilon^j(n) \in \{0, 1\}$ is random reward received by LA j.

In DSA context, the cognitive radio network is considered as the unknown random environment in which the primary channel availability statistics are unknown to the SUs. In this context, the SUs are referred to the learning automata, adaptive decision making agents. The SUs try to learn the optimum channel selection through interaction with the random environment.

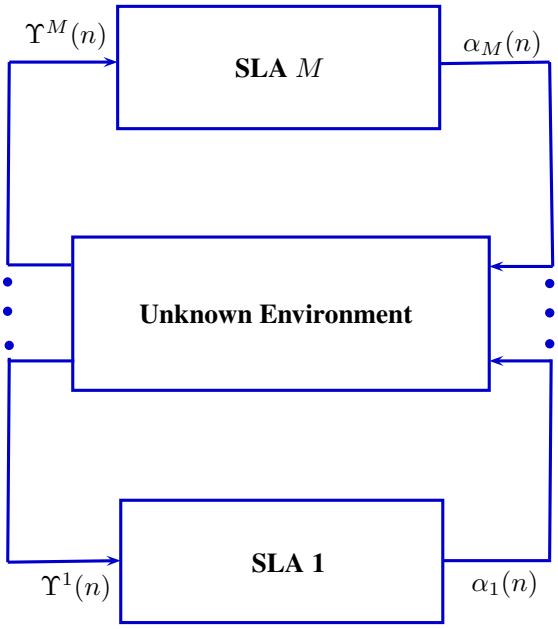

Figure 2.2: Multiple LA.

2.3.3 Related Work

Initial concept of learning automata was introduced in [70, 71]. Early learning automata models were developed in mathematical psychology [64]. Several studies have been conducted in area of LA afterwards [72–77]. SLA has been applied into a variety of areas [78–87]. [84] introduces a SLA-based multimodal searching technique with changing number of actions. In [85], learning automata has been applied in pattern recognition problem. [86] proposes a SLA-based intelligent controller design for an automated vehicle. In their design, using the data obtained from on-board sensors, they consider two automata (one for lateral actions, one for longitudinal actions) to learn the best possible action to avoid collisions. Unlike adaptive control methods or expert systems, their proposed design system is capable of working in unmodeled

stochastic environments.

In [87], the SLA has been incorporated in designing probabilistic power adaptation algorithms. [87] focuses on solutions to distributed discrete power control by proposing two algorithms: 1. Discrete Learning Power Control-I (DSLPC-I) which has exactly the same action selection probability update rule as in [88], and 2. Discrete Learning Power Control-II (DSLPC-II) in which a new action selection probability update rule is applied. In [89], stochastic automata rate adaptation (SARA) algorithm is proposed which implements the SLA in context of the rate adaptation. In this context, the wireless channels are considered as an unknown environment and the transmission rates are the actions. [90] investigates the decentralized multimode precoding strategy selection for MIMO MAC through a game theoretic approach. To achieve the Nash Equilibrium, this work proposes a learning automata based decentralized algorithm. [91] applies SLA in capacity assignment (CA) problem. The goal of the CA problem is finding the best possible set of capacities for the links in a prioritized network. The assigned set of capacities should satisfy the traffic requirements while minimizing the cost. In [92], distributed channel selection in an opportunistic spectrum access (OSA) system is formulated as a repeated game using learning automata approach. [93] has recently proposed an SLA-based algorithm for distributed access. The SLA-based policy proposed in [93] relies on an direct learning automata[3] called Linear Reward-Inaction algorithm [64]. This algorithm does not consider learning of the channel availabilities at the SUs. There, a different mechanism for channel selection and collision resolution is designed. Simulation results also show the effectiveness of our proposed policy compared with this policy in various types of mean availabilities

[3]A direct learning automata is a learning automata in which the environment model is not used in the learning algorithm.

across primary channels. In [94], application of indirect learning automata[4] method in zero-sum game with identical payoff is investigated. The proposed algorithm in [94] is an extension of the well-known Pursuit learning algorithm to the decentralized version.

[4] In an indirect learning automata, the environment model is used in the learning algorithm.

Chapter 3

Dynamic Spectrum Access via Channel-Aware Heterogeneous Multi-channel Auction with Distributed Learning

3.1 Introduction

Designing dynamic spectrum access mechanisms for efficient utilization of the spectrum is one of the main challenges faced in building cognitive radio networks. A hierarchical cognitive radio network consists of a primary network where primary users are licensed to use the network spectrum, and secondary users (SUs) who opportunistically use the idle channels in the primary network unoccupied by the licensed users. The channel availability statistics of the primary network are typically unknown to the SUs. Through limited spectrum sensing, the SUs search for idle channels and make decisions for channel access. The dynamic access mechanism can be designed in a coordinated, distributed or hybrid fashion. Designing a policy or mechanism for spectrum access among SUs involves key challenges in two aspects: One is on the

learning of primary channel conditions by SUs, *i.e.,* how to provide efficient online learning of the primary channel availability statistics based on the sensing observation history. The other is on the handling of access among SUs. The latter not only includes how to provide an effective mechanism to resolve collisions among SUs, but also how to effectively explore the link conditions of the SUs themselves for opportunistic access. All the above issues directly impact the SUs' throughputs.

Consider a cognitive radio network with N independent primary channels and M SUs. Distributed design for spectrum access is desirable to reduce communication overhead and/or delay incurred. Several decentralized learning and access policies have been recently developed [9, 10, 20–23] by formulating the problem as a decentralized multi-armed bandit (MAB) problem [39], *i.e.,* to select the best M out of N channels for access in a distributed manner. In these policies, the unknown mean availability of each channel is estimated through a learning process based on each SU's own sensing observation history. Relying on the estimated mean channel availabilities, each policy designs a different mechanism to avoid or resolve collisions among SUs for their access to the M most available primary channels.

Different from MAB formulations, auction design has recently attracted interests for dynamic spectrum access in cognitive radio networks [13–19]. Treating available channels as objects and SUs as bidders, the channel selection or assignment can be made through an auctioning process. Typically, an auctioneer (or coordinator) is required in such an auction. Compared with the contention-based distributed policies described earlier, it is a collision-free approach for spectrum access among SUs. Several challenges are faced in designing an auction for dynamic access: Each SU needs to decide which primary channels to bid for. Since the number of primary

channels N can be large, and each channel has different loading statistics, bidding for all the primary channels may be undesirable in terms of both large communication overhead and poor throughput due to selecting an often occupied channel. Then, the question is which channel or a subset of channels each SU should bid for, especially when channel availability statistics are unknown to the SUs. Moreover, each SU's throughput directly links to its own instantaneous link condition over the selected primary channel. Although important, such a secondary link condition is typically ignored in the existing distributed policies mentioned above. However, to maximize the SUs' throughputs, both primary channel load and secondary link condition should be taken into account in the auction process for channel selections. In addition, each SU intends to maximize its own payoff through bidding. Thus, it is desirable for the auction mechanism to possess some nice properties for individual performance guarantee.

3.1.1 Contributions

In this chapter, we consider an auction-based approach for spectrum access among SUs and try to address the challenges mentioned above in our auction mechanism design. Our design incorporates distributed online learning of the primary channel availability statistics and explores the secondary instantaneous link condition for multi-user multi-channel diversity gain to improve each SU's throughput. Specifically, we consider primary channels with heterogeneous availability statistics. Through auctioning, each SU obtains a channel and accesses it if idle. Viewing this problem as M bidders bidding for N heterogeneous objects, we adopt a unit demand (UD) auction, known as Demange-Gale-Sotomayor (DGS) auction [8], to determine channel

selection of each SU based on its instantaneous rate over each channel.

To address some main disadvantages faced in the UD auction in this problem, we further propose a learning-based unit demand (LBUD) auction. Each SU only bids for the M most available channels which are learned by each SU distributively using its own sensing history. Note that with unknown primary channel statistics, selecting which set of M channels to bid is highly nontrivial. Our proposed LBUD incorporates the design of distributed learning of primary channel availability at each SU. Thus, choosing which M channels to bid is at the same time making a trade-off between exploration and exploitation for learning. Compared with the UD auction, the proposed LBUD auction not only significantly reduces the required communication overhead, but also improves the throughput performance by avoiding selecting channels with low availability. We further show that the LBUD auction preserves the strong property of the DGS auction, *i.e.,* it is *dominant strategy incentive compatible* (DSIC) (the definition of DSIC is given in Section 3.4.2). This means that each SU will achieve its maximum payoff by bidding truthfully using SU's own instantaneous rate, regardless of whether other SUs bid truthfully or not. In truthful bidding, each SU's bid should reflect its valuation truthfully. In other words, its bid equals to its valuation.

For both UD and LBUD auctions, an iterative procedure is used to determine the channel assignment to each SU. To improve the convergence speed of the iterative procedure, instead of fixed price increment, we further propose an adaptive price increment algorithm to determine price increment in each iteration. Simulations show the effectiveness of our proposed auction mechanism in throughput gain over other existing policies by bidding only selective primary channels and exploring instantaneous

secondary link conditions.

To the best of our knowledge, we are the first to apply the DGS auction designed for bidding heterogeneous objects with unit demand to the distributed dynamic access design. Furthermore, our proposed LBUD auction incorporates distributed learning of the primary channels into the auction mechanism to explore both channel availability statistics and instantaneous link gain of SUs, in order to maximize SUs' throughputs. Such a joint consideration of both primary channel and secondary link is not considered in either existing decentralized MAB policies or auction-based access mechanisms.

3.2 Network Model

We consider a slotted primary network consisting of N orthogonal radio channels available to the licensed primary users. A secondary network with M secondary users search and compete for the instantaneous idle channels among these N channels. Denote $X_i(n) \in \{0, 1\}$ as the availability state of the primary channel i at slot n, with $X_i(n) = 1$ as the channel i being available, and it is 0 otherwise. We assume the channel i's availability state $X_i(n)$ evolves as an i.i.d. Bernoulli random process over time, with $X_i(n) \sim \text{Bernoulli}(\theta_i)$, where $\theta_i = \text{E}[X_i(n)]$, $\forall n$. Let $\boldsymbol{\theta} \triangleq [\theta_1, \cdots, \theta_N]^T$ denote the channel availability vector. We assume elements in $\boldsymbol{\theta}$ are all distinct and unknown to the SUs.

At the beginning of each time slot n, each SU j selects a channel to sense and, if available, to access. We assume perfect channel sensing is performed.

Let $h_i^j(n)$ be the channel gain between SU j and its (secondary) destination (*e.g.* base station) over radio channel i at time slot n. Note that channel i indicates the

frequency channel SU j occupies, while $h_i^j(n)$ is the channel gain over the link between the secondary transceiver, which can be measured by SU j. Assuming channel i is available for SU j to access, the corresponding instantaneous achievable rate of SU j is $R_i^j(n) = \log\left(1 + P_j|h_i^j(n)|^2/\sigma^2\right)$ with P_j and σ^2 being SU j's transmit power and its receiver noise variance, respectively. We assume perfect knowledge of $R_i^j(n)$, $\forall i$, at each SU j. There may be different ways for each SU to obtain its rate in practice. Without causing any interference to primary users, an SU may use the (delayed) rate information obtained from the most recent channel access. Note that, if SUs are bidding only among a subgroup of channels, as what we will propose later in the paper, the delay may be short. In our work, we idealize it to be instantaneous rate. Other short channel probing design for SUs may be possible at the beginning of each time slot to obtain a coarse estimate of its rate, provided the interference to the primary users is kept below a tolerate level.

The expected throughput for the secondary network, under a given access mechanism, is given by

$$\begin{aligned}\mathcal{T}(n) &= \frac{1}{n}\sum_{j=1}^{M}\sum_{i=1}^{N}\mathrm{E}\left[\sum_{k\in\mathcal{I}_i^j(n)} X_i(k)R_i^j(k)\right] \\ &= \frac{1}{n}\sum_{j=1}^{M}\sum_{i=1}^{N}\theta_i\mathrm{E}\left[\sum_{k\in\mathcal{I}_i^j(n)} R_i^j(k)\right], \end{aligned} \tag{3.2.1}$$

where $\mathcal{I}_i^j(n)$ denotes the set of time slots up to the current time slot n that SU j has been the sole user of channel i. In other words, $\mathcal{I}_i^j(n)$ only contains the set of time slots up to the current time slot n that SU j does not collide with other SUs on channel i. Therefore, the collision among the secondary users is reflected in $\mathcal{I}_i^j(n)$ in the expected throughput in (3.2.1). Our problem is to design a distributed

online learning and access mechanism among SUs to maximize the secondary network throughput defined above. Different from existing learning and access designs, with channel fading taken into account, our design explores the instantaneous channel gain over (secondary) links to maximize the secondary network throughput.

3.3 Dynamic Access via Multi-channel Auction

A major challenge faced in distributed access among SUs to the primary network is the design of collision avoidance among SUs for their access. We consider an auction-based access mechanism, in which each SU performs online learning of the primary channels individually while their access channel selection is managed by an auctioneer (or coordinator). The benefit of such an auction mechanism is to enable a collision free access.

Let $\mathcal{S}$ denote the set of SUs and $\mathcal{C}$ the set of the primary channels. Consider SUs as the bidders and the primary channels as the objects of the auction. In this section, we first consider each SU can bid for any channel in $\mathcal{C}$. In Section 3.4, we modify our auction mechanism to consider bidding channels in a subset of $\mathcal{C}$. At the beginning of time slot n, SU j sends to the auctioneer a confidential bidding vector of all the primary channels, denoted as $\mathbf{m}^j(n) \triangleq [m_1^j(n), \cdots, m_N^j(n)]^T$, where $m_i^j(n)$ denotes the bid of SU j for channel i. If SU j decides not to participate in bidding of channel i, then $m_i^j(n) = 0$. We also define $\mathbf{m}^{-j}(n)$ as the bidding vectors of SU j's opponents, *i.e.,* $\mathbf{m}^{-j}(n) = \{\mathbf{m}^k(n) | k \in \mathcal{S} \backslash j\}$. Based on the bids from SUs, the auctioneer will allocate a channel to each SU.

Note that since the mean availability statistics θ_i's are distinct, these primary channels that the SUs bid for are considered as the heterogeneous type. Thus, the

problem is essentially the bidding of multiple heterogenous objects with unit demand. This is considered as a *unit demand auction*. In economics, DGS auction [8] was first proposed to handle such a scenario. It is a generalization of the second-price auction [31] which deals with multiple bidders bidding for a single object. The DGS auction preserves some nice properties of the second-price auction. It is a *weakly dominant strategy* which leads to a dominant strategy equilibrium in which the payoff of each bidder is maximized regardless of other bidders' strategies. In game theory, a player's strategy is called a weakly dominant strategy if it is at least as good as any other strategy for that player irrespective of what other players' strategies are. Note that the dominant strategy equilibrium is a Nash equilibrium, but not vice versa. In addition, the DGS auction reaches the minimum price equilibrium [8].

Let $A_i^j(n) \in \{0, 1\}$ indicates the allocation of channel i to SU j at time slot n, with the value of 1 if channel i is assigned to SU j. Each channel can be assigned to at most one SU, and each SU can be assigned at most one channel. Denote $\mathbf{A}_i(n)$ the allocation of channel i to SUs at time slot n and $\mathbf{A}^j(n)$ the channel allocation for SU j at time slot n, respectively. They are given by

$$\mathbf{A}_i(n) \triangleq \{A_i^1(n), \cdots, A_i^M(n) : \sum_{j=1}^{M} A_i^j(n) \leq 1\} \tag{3.3.1}$$

$$\mathbf{A}^j(n) \triangleq \{A_1^j(n), \cdots, A_N^j(n) : \sum_{i=1}^{N} A_i^j(n) \leq 1\} \tag{3.3.2}$$

In addition, if an SU does not bid for a channel, it will not be assigned to that channel, *i.e.,* $A_i^j(n)|\{m_i^j(n) = 0\} = 0$. A reservation price $P_{\min,i}$ is given to each channel i, indicating the minimum price the auctioneer accepts for a specific channel. Let $\mathbf{P}_{\min} \triangleq [P_{\min,1}, \cdots, P_{\min,N}]^T$. The auction mechanism uses an iterative procedure to determine the channel assignment for each SU. Let $P_i^l(n)$ denote the price for

channel i at time slot n at iteration l. Let $\mathbf{P}^l(n) \triangleq [P_1^l(n), \cdots, P_N^l(n)]^T$.

There are three terms used in the UD auction, *demand* set, *overdemanded* set and *minimal overdemanded* set. We first provide their definitions below.

Definition 1 ([8]). Demand set – demand set *for each SU j, denoted as $\mathcal{D}^j(\mathbf{P}^l(n))$. It is defined as the set of channels that give the current maximal payoff for SU j, i.e.,*

$$\mathcal{D}^j(\mathbf{P}^l(n)) = \left\{ \arg\max_{i \in \mathcal{C}} (R_i^j(n) - P_i^l(n)) \right\}. \tag{3.3.3}$$

Definition 2 ([8]). Overdemanded set – *Define the set of SUs as* demanders *of $\mathcal{D}^j(\mathbf{P}^l(n))$ as*

$$\mathcal{B}(\mathcal{D}^j(\mathbf{P}^l(n))) = \{k : \mathcal{D}^j(\mathbf{P}^l(n)) \cap \mathcal{D}^k(\mathbf{P}^l(n)) \neq \emptyset, \forall k \in \mathcal{S}\}.$$

Define the set of SUs as exclusive demanders *for $\mathcal{D}^j(\mathbf{P}^l(n))$ as*

$$\mathcal{B}^E(\mathcal{D}^j(\mathbf{P}^l(n))) = \{k : \mathcal{D}^k(\mathbf{P}^l(n)) \subseteq \mathcal{D}^j(\mathbf{P}^l(n)), \forall k \in \mathcal{S}\}. \tag{3.3.4}$$

We call the set (of channels) $\mathcal{D}^j(\mathbf{P}^l(n))$ being overdemanded *at price $\mathbf{P}^l(n)$ if* [1]

$$\mathcal{D}^j(\mathbf{P}^l(n)) \subset \mathcal{C} \text{ and } |\mathcal{B}^E(\mathcal{D}^j(\mathbf{P}^l(n))| > |\mathcal{D}^j(\mathbf{P}^l(n))|. \tag{3.3.5}$$

In other words, the set of channels is overdemanded, if there are more SUs, whose highest payoff channels are all in this set, than the number of channels in the set.

Definition 3 ([8]). *Minimal Overdemanded set – An overdemanded set $\mathcal{D} \in \mathcal{O}$ is called a minimal overdemanded set if $\mathcal{D}' \notin \mathcal{O}$, $\forall \mathcal{D}' \subset \mathcal{D}$.*

We summarize the UD auction mechanism for the allocation decision:

[1] *The set $\mathcal{D}^j(\mathbf{P}^l(n))$ is called weakly overdemanded if "$>$" in* (3.3.5) *is replaced by "$\geq$".*

1. The auctioneer initializes the price to the reservation price for each channel: $\mathbf{P}^0(n) = \mathbf{P}_{\min}$;

2. For each SU j, $j \in \mathcal{S}$, it observes its current valuation of each channel, *i.e.,* the instantaneous rate $R_i^j(n), \forall i$. Since bidding truthfully is a weakly dominant strategy in the UD auction, we set the bid to be the valuation of the channel, $m_i^j(n) = R_i^j(n)$. SU j then sends $\mathbf{m}^j(n)$ to the auctioneer;

3. The auctioneer obtains the *demand set* for each SU j, $\mathcal{D}^j(\mathbf{P}^l(n))$ as in (3.3.3).

4. Let $\boldsymbol{\mathcal{D}}(\mathbf{P}^l(n)) \triangleq \{\mathcal{D}^1(\mathbf{P}^l(n)), \cdots, \mathcal{D}^M(\mathbf{P}^l(n))\}$. The auctioneer follows an iterative procedure to check whether there is any *overdemanded set* among the demand sets in $\boldsymbol{\mathcal{D}}(\mathbf{P}^l(n))$:

   4.1) *If there is no overdemanded set of channels*: The channel allocated to SU j is given as $A_i^j(n) = 1$, where $i \in \mathcal{D}^j(\mathbf{P}^l(n))$ is selected randomly for $|\mathcal{D}^j(\mathbf{P}^l(n))| > 1$,[2] and $A_{i^-}^j(n) = 0$, for $i^- \in \mathcal{C}\backslash\{i\}$. The allocation process is completed and terminated. The final price on channel i, defined by $P_i(n)$, is given by

   $$P_i(n) = P_i^{l+1}(n).$$

   4.2) *If there are overdemanded sets of channels*:

   4.2a) Let $\mathcal{S}^o$ be the set of SUs whose $\mathcal{D}^j(\mathbf{P}^l(n))$ is an overdemanded set. The auctioneer collects all the overdemanded sets into a set $\mathcal{O}$

   $$\mathcal{O} = \{\mathcal{D}^j(\mathbf{P}^l(n)), \forall j \in \mathcal{S}^o\}. \tag{3.3.6}$$

[2] $|\cdot|$ denotes the cardinality of a set.

4.2b) The auctioneer finds a *minimal overdemanded set* $\mathcal{D}^{\min}(\mathbf{P}^l(n))$ from $\mathcal{O}$,[3] and updates the price vector $P_i^l(n)$, for $i \in \mathcal{D}^{\min}(\mathbf{P}^l(n))$:

$$P_i^{l+1}(n) = P_i^l(n) + \Delta P_{n,l} \tag{3.3.7}$$

where $\Delta P_{n,l}$ is the price increment at iteration l and time slot n. Return to Step 3.

Remark 1: The minimal overdemanded set might not be unique. Therefore, if there is more than one over demanded set, then one of them is randomly selected by the auctioneer.

Remark 2: It has been shown that the above UD auction mechanism leads to a minimal price equilibrium [8]. That is, let $\mathbf{P}^*(n)$ be the price obtained at the end of the auction, and $\mathbf{q}(n)$ be any other competitive price vector at time slot n. Then, $\mathbf{P}^*(n) \leq \mathbf{q}(n)$. In addition, since the auction is shown to be weakly dominant strategy, each SU obtains its maximal payoff regardless of other bidders' strategies.

Remark 3: In the above UD auction mechanism, each SU uses its instantaneous rate (of the secondary link) to bid; thus, the channel assignment from the auction depends on the instantaneous link condition for each SU on the primary channels. Therefore, the channel assignment is opportunistic, and the resulting throughput at the secondary network gains from such an opportunistic allocation, in addition to be collision free.

[3]There can be more than one minimal overdemanded set. In this case, the auctioneer randomly selects one.

3.4 Multi-channel Auction via Distributed Learning

In the UD auction mechanism described in the previous section, each SU bids for all N channels. We are interested in the scenario where $M < N$. This scenario arises in broadband spectrum access where there are a large number of primary channels for consideration. This scenario is particularly suitable for the cognitive radio network where the SUs may not be restricted to a certain frequency band and can search among a large set of channels. In this case, there are two drawbacks to this approach: First, there are a total of MN bids submitted to the auctioneer which incurs a large communication overhead. Second, the payoff used in determining the channel allocation only reflects each SU's instantaneous rate of its link on a channel, but does not take into account the different mean channel availabilities among the primary channels. The latter could result in more throughput loss by selecting a primary channel that is less available. To overcome these drawbacks, we propose an adaptive auction mechanism, named the LBUD auction. In this auction, each SU j will adaptively choose the best M channels to bid.

Since the channel mean availability θ_i of each channel i is unknown to the SUs, to determine the most available M channels, each SU learns $\boldsymbol{\theta}$ distributively from its own sensing outcome and history over time. Since one of the main motivations is to reduce the overhead, we consider distributed learning of M-best channels at each SU. A centralized learning would require each SU to submit its sensing results to the auctioneer and obtain the estimated M-best channels from the auctioneer every time slot. Let $T_i^j(n)$ denote the number of times that the SU j senses channel i up to time

slot n. For SU j selecting channel i to sense at time slot n, it records the value of $X_i(n)$ as $X_i^j(T_i^j(n))$. For SU j, its sensing observation history of channel i up to time slot n is denoted by $\mathbf{X}_i^j(n) \triangleq [X_i^j(1), \cdots, X_i^j(T_i^j(n))]^T$. SU j estimates θ_i of channel i at time slot n using the sample mean of the observations from $\mathbf{X}_i^j(n)$ as

$$\hat{\theta}_i^j(T_i^j(n)) = \frac{1}{T_i^j(n)} \sum_{k=1}^{T_i^j(n)} X_i^j(k). \tag{3.4.1}$$

The online learning algorithm, upper-confidence-bound1 (UCB1) [44], is a sample-mean based index policy to learn and access N channels in a single user scenario. Through efficient exploration-exploitation trade-off, the UCB1 algorithm has been shown to be order-optimal in terms of the learning rate over time. Existing decentralized policies [9,10,20] have extended the UCB1 algorithm to a multi-user scenario with decentralized learning at each SU. In the UCB1 algorithm, each SU j ranks channel i at time slot n based on an index, $I_i^j(n)$, defined as

$$I_i^j(n) \triangleq \hat{\theta}_i^j(T_i^j(n)) + \sqrt{\frac{2\log n}{T_i^j(n)}}. \tag{3.4.2}$$

The SU computes the index vector $\mathbf{I}^j(n) \triangleq [I_1^j(n), \cdots, I_N^j(n)]^T$ based on its own sensing observation history. Note that the two terms in (3.4.2) capture the exploration and exploitation trade-off in learning. The sample mean for estimated channel availably in the first term corresponds to exploitation, while the second term is used for exploration which adds weights to those channels that are not sensed often. Thus the trade-off is between choosing a channel with a high estimated availability for immediate throughput maximization and choosing another channel to obtain an improved estimate of its availability.

Define the M-best channels as those channels whose θ_i's are among the M highest ones. We also define the estimated M-best channels by SU j as those channels

whose indexes $I_i^j(n)$'s in (3.4.2) by SU j are among the top M-ranked. Thus, the estimated M-best channels reflects the exploration-exploitation trade-off captured by $I_i^j(n)$ in (3.4.2) in the underlying learning. Let $\mathcal{C}_M^j(n)$ denote the set of indexes of the estimated M-best channels for SU j at time slot n, *i.e.,*

$$\mathcal{C}_M^j(n) = \left\{ i : I_i^j(n) \in \left\{ I_{(1)}^j(n), \cdots, I_{(M)}^j(n) \right\} \right\}, \tag{3.4.3}$$

where $\{I_{(i)}^j(\cdot)\}$ is the ordered statistics of $\{I_i^j(\cdot)\}$ with $I_{(1)}^j(\cdot) > \cdots > I_{(N)}^j(\cdot)$. At each time slot n, SU j updates its estimated M-best channel set $\mathcal{C}_M^j(n)$, and form the bidding vector for these channels: $\mathbf{m}_M^j(n) = [m_{k_1}^j(n), \cdots, m_{k_M}^j(n)]^T$, where $k_i \in \mathcal{C}_M^j(n)$. The auctioneer performs an auction-based allocation using these bidding vectors from SUs. The demand set $\mathcal{D}_M^j(\mathbf{P}^l(n))$ for SU j in this case is given by

$$\mathcal{D}_M^j(\mathbf{P}^l(n)) = \left\{ \arg\max_{i \in \mathcal{C}_M^j(n)} (m_i^j(n) - P_i^l(n)) \right\}. \tag{3.4.4}$$

We will show in Section 3.4.2 that using the true valuation of a channel as the bid, *i.e.,* the instantaneous rate on the channel, will lead to the maximum payoffs among SUs. Therefore, we set $m_i^j(n) = R_i^j(n)$ in the following. Using $\mathcal{D}_M^j(\mathbf{P})$, we design an iterative procedure to determine the channel allocation among M SUs. The LBUD auction mechanism is described in Algorithm 1 (see Fig. 3.1).

Note that, in terms of overhead, in the LBUD auction, each SU only submits M bids along with the channel index set $\mathcal{C}_M^j$. The total overhead is M^2 bids plus $M \log_2 M$ bits, in contrast to MN bids in the UD auction. For accessing broadband spectrum with $N \gg M$, the reduction in communication overhead is significant. Performance-wise, each SU only bids among its estimated M-best channels, and at the same time, the channel allocation is based on the instantaneous (secondary) link condition on each channel. Thus, the LBUD auction is designed to 1) ensure good

channel selection in the mean sense; 2) enjoy the gain from opportunistic channel selection. Note that existing distributed learning and access policies [9, 10, 20] for spectrum access only ensure good channel selection in the mean sense without considering instantaneous channel condition of SUs' communication links.

For the LBUD auction, the winning channel will be selected only among those highly available channels. When the primary channel mean availability values in $\boldsymbol{\theta}$ are relatively spread, this will result in the LBUD auction outperforming the UD auction. This is because bidding only among the M-best channels avoids SUs to access the channels which are less likely to be available (as in the UD auction), even though the SU's instantaneous rates over these channels are high. However, when the values in $\boldsymbol{\theta}$ are close, the UD auction may outperform the LBUD auction, due to the *multi-channel diversity gain* from the opportunistic selection of M out of N channels. To cover a broad range of the distribution of primary channel availability statistics $\boldsymbol{\theta}$, in practice, we should consider both mechanisms to adapt to different types of traffic conditions over channels in the primary network. It should be mentioned that the LBUD auction becomes the UD auction when $M \geq N$. However, for $M < N$, there is a nontrivial learning process of the M-best channels involved. In this case, the LBUD auction improves the throughput performance and reduces the communication overhead than the UD auction.

3.4.1 Adaptive Algorithm for Price Increment $\Delta P_{n,l}$

The iterative procedure in both the UD and the LBUD auctions requires the price update with the price increment $\Delta P_{n,l}$, as shown in (3.3.7) and (3.4.5). Setting the appropriate price increment $\Delta P_{n,l}$ is important as it directly affects the convergence

behavior of the iterative procedure for the auction. If the increment is too small, the overdemanded set will not change over several iterations, resulting in slow convergence. However, if the increment is too large, the iterative procedure may not guarantee to converge. Note that the DGS auction is originally proposed for integer valuations and prices. Due to this, the price increment in the iteration procedure is fixed to $\Delta P_{n,l} = 1$, *i.e.,* the minimum possible difference of two non-identical valuations. The convergence has been shown with this unit increment [8]. In our case, the valuations (instantaneous rates) are real numbers. In this case, we can similarly set $\Delta P_{n,l}$ to be the minimum difference of instantaneous rates of all channels that each SU bids for.

Let $\tilde{\mathcal{C}}^j(n)$ be the primary channel set considered by SU j: for DGS, $\tilde{\mathcal{C}}^j(n) = \mathcal{C}$, and for LBUD, $\tilde{\mathcal{C}}^j(n) = \mathcal{C}_M^j(n)$. Then, the price increment $\Delta P_{n,l}$ is set as

$$\begin{aligned} \Delta P_{n,l} &= \min\left\{\left|R_i^j(n) - R_m^j(n)\right| : i, m \in \tilde{\mathcal{C}}^j(n), i \neq m, \forall j\right\} \\ &\triangleq \Delta P_n^b \end{aligned} \tag{3.4.6}$$

where ΔP_n^b is denoted as the *baseline price adjustment*.

Note that, by this baseline price increment method, $\Delta P_{n,l}$ is fixed for each auction procedure, but varies from slot to slot. Using the price increment suggested in (3.4.6) may lead to slow convergence. To improve the convergence rate, we propose an adaptive algorithm which updates $\Delta P_{n,l}$ adaptively in each iteration l. It is described as follows (for the auctioneer, at time slot n and iteration l):

Let $\rho_i^j(n,l)$ denote the current payoff of SU j over channel i at time slot n at iteration l, *i.e.,*

$$\rho_i^j(n,l) = R_i^j(n) - P_i^l(n). \tag{3.4.7}$$

The auctioneer obtains $\rho_i^j(n,l)$ for each SU $j \in \mathcal{S}$ and channel i. Let k_1^j and k_2^j denote the channels with the current maximum and second maximum payoff for SU j at time slot n and iteration l, respectively[4]. They are obtained by

$$k_1^j = \arg\max_{i \in \tilde{\mathcal{C}}^j(n)} \rho_i^j(n,l). \tag{3.4.8}$$

$$k_2^j = \arg\max\{\rho_i^j(n,l) : \rho_i^j(n,l) \neq \rho_{k_1^j}^j(n,l),\ i \in \tilde{\mathcal{C}}^j(n)\}. \tag{3.4.9}$$

For each SU j, let $\Delta\rho^j(n,l)$ be the payoff difference over the two channels k_1^j and k_2^j, given by

$$\Delta\rho^j(n,l) = \left|\rho_{k_1^j}^j(n,l) - \rho_{k_2^j}^j(n,l)\right|. \tag{3.4.10}$$

The auctioneer updates the price increment $\Delta P_{n,l}$ as

$$\Delta P_{n,l} = \min_{1 \leq j \leq M} \Delta\rho^j(n,l). \tag{3.4.11}$$

The convergence of the iterative procedure with the above adaptive price increment is shown in the following proposition.

Proposition 1. *Under the proposed adaptive price increment algorithm, the iterative procedure in both the UD and the LBUD auctions converges to the same channel assignment outcome as that of the baseline price adjustment method.*

Proof. See Appendix A.1.

Remark 1: It can be seen that the price increment $\Delta P(n,l)$ is adaptively determined based on the gap of current top payoffs $\{\rho_i^j(n,l)\}$ over primary channels among SUs. Intuitively, the adaptive price increment algorithm is designed to skip

[4]If there are multiple channels having the same maximum (or second maximum) payoffs, randomly select one channel as m_1^j (or m_2^j).

those iterations where there is no change to the overdemanded sets and minimal overdemanded sets among SUs, and thus they will not affect the outcome of channel assignment result. By doing so, the algorithm avoids those "null" iterations resulting in fewer iterations to reach the same channel assignment solution.

Remark 2: As indicated in Proposition 1, the adaptive price increment results in the same allocation as that of the baseline price increment. Therefore, our proposed adaptive price increment algorithm does not affect the allocation efficiency of the auction mechanism (UD or LBUD).

Such an adaptive increment avoids unnecessary iterations and expedites the convergence to the final channel assignment for each SU at the auctioneer. In Section 3.5, through simulations, we show that the proposed adaptive price increment algorithm substantially improves the convergence rates of both the DGS and the LBUD algorithms.

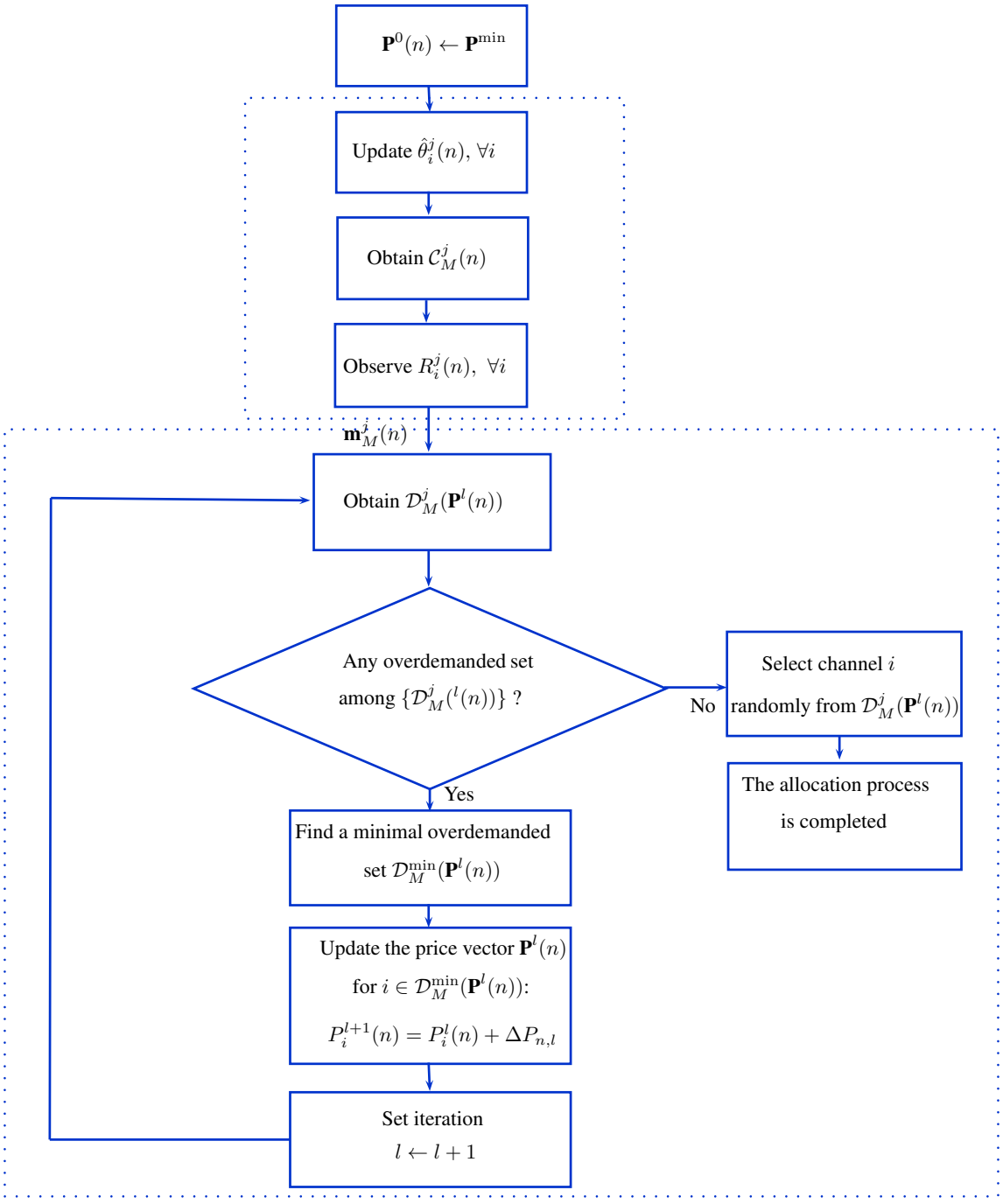

Figure 3.1: : Learning-Based Unit Demand (LBUD) Auction

Algorithm 1 : Learning-Based Unit Demand (LBUD) Auction

1) **Input**:

   n: Current time slot
   l: Current iteration

2) **Init**: Set the price vector $\mathbf{P}^0(n) \leftarrow \mathbf{P}_{\min}$

3) SU j updates $\hat{\theta}_i^j(n)$ using (3.4.1), $\forall i$, and obtain $\mathcal{C}_M^j(n)$ using (5.3.1).

4) SU j observes its current link rate $R_i^j(n)$, $\forall i$ and sends a confidential bidding vector $\mathbf{m}_M^j(n)$.

5) The auctioneer obtains the demand set $\mathcal{D}_M^j(\mathbf{P}^l(n))$ for SU j using (3.4.4).

6) The auctioneer checks whether there is any overdemanded sets among the demand sets $\{\mathcal{D}_M^j(\mathbf{P}^l(n))\}$:

   a) The auctioneer obtains the exclusive demanders $\mathcal{B}^E(\mathcal{D}_M^j(\mathbf{P}(n)))$ of $\mathcal{D}_M^j(\mathbf{P}(n))$ as in (3.3.4), $\forall j$.

   b) The auctioneer checks whether $\mathcal{D}_M^j(\mathbf{P}^l(n))$ is an overdemanded set or not as in (3.3.5), $\forall j$:

      If there is no overdemanded set:

      The channel allocated to SU j is given as $A_i^j(n) = 1$, where $i \in \mathcal{D}_M^j(\mathbf{P}^l(n))$ is selected randomly for $|\mathcal{D}_M^j(\mathbf{P}^l(n))| > 1$ and $A_{i^-}^j(n) = 0$, for $i^- \in \mathcal{C}\backslash\{i\}$.
      The allocation process is completed and terminated.
      The final price on channel i, $P_i(n)$, is given by

$$P_i(n) = P_i^{l+1}(n)$$

      If there are overdemanded sets:

      i) The auctioneer forms Set $\mathcal{O}$ as in (3.3.6).

      ii) The auctioneer finds a minimal overdemanded set $\mathcal{D}_M^{\min}(\mathbf{P}^l(n))$ from $\mathcal{O}$, and updates the price vector $\mathbf{P}^l(n)$, for $i \in \mathcal{D}_M^{\min}(\mathbf{P}^l(n))$, as

$$P_i^{l+1}(n) = P_i^l(n) + \Delta P_{n,l}. \tag{3.4.5}$$

      iii) Set iteration $l \leftarrow l + 1$; return to Step 5.

Algorithm 2 : Adaptive Price Increment Algorithm
($\Delta P_{n,l}$ at the lth iteration)

1) **Input**:
   n: Current time slot
   l: Current iteration
2) Obtain $\rho_i^j(n,l)$ as in (3.4.7) for each SU $j \in \mathcal{S}$ and channel i.
3) Obtain k_1^j as in (3.4.8) and k_2^j as in (3.4.9).
4) Obtain $\Delta\rho^j(n,l)$ as in (3.4.10).
5) Update the price increment $\Delta P_{n,l}$ as in (3.4.11).

3.4.2 Property of the LBUD Auction

An auction mechanism is said to be *incentive compatible* if all bidders will receive the maximum payoffs when their bids reflect their valuations truthfully [95]. Furthermore, a strategy is called *dominant strategy incentive compatible*, *i.e.,* DSIC, where each bidder achieves its maximum payoff by bidding truthfully irrespective of whether the other bidders bid truthfully or not [95]. Under a dominant strategy, the bidders of this auction do not need to collect or analyze any information about the status or intentions of their competing bidders.

Bidding truthfully simplifies decision making of the SUs in an auction. It is because by bidding truthfully, the secondary users need to know only their own valuations and therefore, they do not depend on knowledge of the other bidders and their distribution of possible values. It has been shown in [8] that bidding truthfully is a dominant strategy in the UD auction. Thus, the UD auction described in Section 3.3 is dominant strategy incentive compatible. We now show that the same property also holds for the proposed LBUD auction as the underlying learning of primary channels by each SU improves.

Proposition 2. *In the long run, the proposed LBUD auction is DSIC.*

Proof. See Appendix A.2.

3.4.3 The UD and the LBUD Auctions: Complexity vs. Overhead

In the UD auction mechanism, SUs bid for all N primary channels. A total of MN bids need to be sent to the auctioneer. This can result in a large communication overhead for large N. Unlike the UD auction, in the LBUD auction mechanism, each SU learns the primary channel occupation statistics and bids for the top M channels that are considered by the SU to be the most available at the current time. Thus, bidding adaptively in the LBUD auction mechanism reduces communication overhead from MN bids in UD auction to M^2 bids. The reduction can be particularly substantial in the broadband primary network environment where $M \ll N$. Thus, the LBUD auction mechanism not only improves the throughput performance when the primary channels have dissimilar availability statistics, but also reduces communication overhead.

In terms of the assignment complexity at the auctioneer, the adaptive price increment algorithm in Section 3.4.1 improves the convergence speed of both the UD and the LBUD auctions. Between the two auction mechanism, our simulations show that the LBUD auction mechanism takes more iterations to converge to the final channel assignment, as compared with that of the UD auction. To see why this is the case, note that the channels each SU can bid for are restricted to its estimated M-best channels. This increases the chance that SUs select the same channels during the iterative procedure. This may lead to a more likely event of having an overdemanded set (of channels), and as a result, slower convergence as compared with that for the

UD auction.

3.5 Simulation Results

In this section, we present the simulation results to assess the performances of the adaptive pricing algorithm and our proposed LBUD auction mechanism. We assume M SUs independently searching for idle channels among N primary channels. In each time slot n, the channel availability state $X_i(n)$ for channel i is independently drawn from Bernoulli distribution with mean θ_i, unknown to SUs. We assume i.i.d Rayleigh fading for $h_i^j(n)$ of SU j's link on channel i over time and for different i and j. The average received SNR over the secondary link, denoted as SNR $\triangleq P_j E[|h_i^j(n)|^2]/\sigma^2$, is set to be 8 dB. All simulations are performed for 50 Monte Carlo runs. We list all the case examples of the mean channel availability vector $\boldsymbol{\theta}$ considered in our simulations in Table 5.1. Cases 1 to 4 represent four different types of primary channel traffic loads for $N = 9$ and $M = 4$. Case 1 is a special case where the channels are all available. Case 2 represents a scenario where the average loads across different channels are random. Case 3 represents a case of dissimilar channels, where the loads are evenly spread out across channels. On the contrary, case 4 shows a scenario where the average loads on all channels are similar. The similar examples for $N = 15$ and $M = 6$ are listed as cases 5 to 7.

3.5.1 Adaptive Price Increment

We show the improvement of the convergence speed using the adaptive price increment algorithm for $\Delta P_{n,l}$ proposed in Section 3.4.1 (Algorithm 2). The channel availabilities

are set randomly as case 2. We set the average received SNR over each secondary link to be SNR = 8 dB.

We first consider the integer-valued example, where prices and rates are all integers. In Fig. 3.2, we plot CDF of number of iterations required to reach the final channel assignment under the UD auction for unit price increment ($\Delta P_{n,l} = 1$) and our proposed price increment. The same comparison is also shown for the LBUD auction in Fig. 3.3.

We also consider the non-integer-valued example, where prices and rates are real values. In Fig. 3.4, for the UD auction, we show the CDF of the number of iterations required to reach the final channel assignment under the adaptive price increment algorithm, as compared with that under the fixed baseline price increment in (3.4.6). The same comparison is also shown for the LBUD auction in Fig. 3.5. For both the UD and the LBUD auctions, we observe the substantial improvement of the convergence rate provided by the adaptive price increment algorithm, with the improvement being particularly pronounced in the LBUD auction. The adaptive price increment algorithm typically takes only a few iterations to reach the channel assignment solution. This thus leads to a significant reduction in complexity.

3.5.2 Impact of Learning and Exploiting Secondary Link Gain

Using Instantaneous Rate as Truthful Bidding

The bidding in both the UD and the LBUD auctions is a truthful bidding by using the instantaneous rate each SU has over its own secondary link. To study this effect

N	M	Case	$\boldsymbol{\theta}$
9	4	1	[1, 1, ... , 1]
		2	[0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7]
		3	[0.1, 0.2, ... , 0.9]
		4	[0.71, 0.72, ... , 0.79]
15	6	5	[0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7, 0.1, 0.45, 0.98, 0.56, 0.27, 0.43]
		6	[0.1, 0.15, ... , 0.75, 0.8]
		7	$0.7 + 0.005 \times$ [2, 3, ... , 14, 16, 18]

Table 3.1: Simulation cases of mean channel availability $\boldsymbol{\theta}$.

on the performance, we consider a different bidding, *i.e.,* sample mean bidding, where the bid that each SU submits is the estimated mean channel availability θ_i of each primary channel i. In Fig. 3.6, we compare the average payoff under the truthful bidding and sample mean bidding cases. The average payoff per SU at time slot n, denoted by $S(n)$, is given by

$$S(n) \triangleq \frac{1}{nM} \sum_{j=1}^{M} \sum_{i=1}^{N} \theta_i \mathrm{E} \left[\sum_{k \in \mathcal{I}_i^j(n)} (R_i^j(k) - P_i(k)) \right]. \tag{3.5.1}$$

As it can be seen, for both the UD and the LBUD auctions, a substantial gain in payoff is achieved by bidding channels using the instantaneous secondary link channel gain rather than the sample mean availabilities of the primary channels. Additional gain is achieved by the LBUD auction as compared to the UD auction by further bidding only among the estimated M-best channels.

Multi-user Diversity Gain

We study the effect of instantaneous fading conditions of the secondary links on the performance under the LBUD auction mechanism. To demonstrate this, we first consider an example where all the primary channels have similar load on average. Specifically, we set $N = 15$ and the mean channel availabilities as case 7. Since θ_i's are similar, the primary channel is chosen would not be a factor that affects the throughput performance of an SU. However, the instantaneous fade of the secondary link over different channels does affect the throughput and is exploited in our proposed LBUD auction. This effect can be observed in Fig. 3.7. which shows the average throughput per SU for $M = 2, 4, 6, 8$. We see that the average throughput per SU increases as M increases. The reason is that, in the LBUD auction, each SU will be assigned one of its estimated M-best channels. As M increases, the SU can choose from more channels. Since the secondary fades over these channels are independent, the channel selection can take advantage from the instantaneous fade gains to capture a *multi-user diversity gain*; and this gain increases with M. Note that, since existing access policies do not consider instantaneous secondary fades, such a multi-user diversity gain cannot be achieved.

Gains from Multi-channel Diversity vs. Using the M-best Channels

In Fig. 3.8, we compare the average throughput per SU under the UD and the LBUD auctions, for cases 2 to 4. From different cases of $\boldsymbol{\theta}$ distributions, we see that, there is a trade-off between the gain of selecting channels among those that are less loaded and the gain of multi-channel diversity. In cases 2 and 3, the average loads across channels are relatively spread out. Learning the mean availability of primary channels to avoid

selecting those more loaded inferior channels is important to prevent throughput loss at SUs. Thus, the gain of selecting channels with higher mean channel availability outweighs the loss of multi-channel diversity due to only bidding among the M-best channels, and the LBUD auction outperforms the UD auction. In case 4, the average loads are similar among channels, and choosing different channels will not impact the SUs' throughputs. Instead, being able to choose from more channels will provide a more pronounced multi-channel diversity gain. The gain of choosing among the M-best channels diminishes, and the UD auction outperforms the LBUD auction due to the gain of multi-channel diversity. The experiments are repeated for cases 5 to 7. The results are shown in Fig. 3.9, where the similar trade-offs are observed.

3.5.3 Comparison with Existing Access Policies

We further compare the performances of the UD and the LBUD auctions with that of the existing access policies. Specifically, the ρ^{RAND} policy [9] and the DLF policy [10] are two existing decentralized access policies we are comparing with. Each policy implements the UCB1 algorithm as its underlying learning of the primary channel mean availabilities, and devises different mechanisms for channel selection and collision resolution. They are briefly described below:

- ρ^{RAND} *policy [9]:* Each SU j selects a random rank r_j uniformly from 1 to M. It will then access the channel i whose $I_i^j(n)$ is ranked r_j^{th} in $\mathbf{I}^j(n)$. At time slot n, if a collision occurred in the previous slot, SU j will re-draw r_j; Otherwise, it keeps the previously generated rank r_j for channel selection.

- *DLF policy [10]:* At time slot n, SU j selects the r_j^{th}-rank channel to access among the top M-ranked channels in terms of $\mathbf{I}^j(n)$, where the rank r_j for each

SU is generated in a round robin fashion $r_j = ((j+n) \mod M) + 1$.

The ρ^{RAND} and the DLF policies are distributed with no central auctioneer, thus there are collisions but with less overhead. In addition, the channel selections of these two policies only rely on mean channel statistics but do not utilize the instantaneous channel gains of the SUs.

In addition, for comparison, we consider the Bertsekas auction [30]. The Bertsekas auction provides a solution to the problem of M bidders (with unit demand) bidding among N objects. It is different from the UD auction as it does not consider the bidders incentives and thus is not dominant strategy incentive compatible. In addition, it cannot handle the primary network load condition.

For our comparison purpose, we provide a modified version of the Bertsekas auction which takes into account the primary channel condition. In the original Bertsekas auction, each SU is trying to find two channels with the highest and the second highest payoffs, respectively, among all channels. In the modified version, we let each SU find these two channels among its estimated M-best channels. Similar to the Bertsekas auction, this modified version is different from the LBUD auction as it does not consider the bidders' incentives and therefore is not dominant strategy incentive compatible.

With the default setup parameters, Figs. 3.10 to 3.12 show the average throughput versus time slot n for the distribution of $\boldsymbol{\theta}$ in cases 2 to 4, under the aforementioned access policies. We also compare the performances of the UD and the LBUD auctions with that of the centralized LBUD auction. Unlike the LBUD auction, learning of the primary channel availabilities is a centralized learning in centralized LBUD auction.

In Figs. 3.13 to 3.15, we also compare the average throughput for these schemes with different mean channel availability $\boldsymbol{\theta}$, case 5, case6, and case 7 respectively where $M = 6$ and $N = 15$.

As it can be seen from these figures, both the LBUD and the UD auctions have asymptotically almost the same performance as the centralized MAB policy and outperform the centralized MAB when all channels have similar loads. The LBUD and the UD auctions also substantially outperform all the other access policies. The LBUD auction has the best performance for most cases except for the case when all channels have similar loads.

3.6 Summary

In this chapter, we considered the auction-based approaches for dynamic spectrum access with unknown primary channel availability statistics to the SUs. Assuming the primary channels are with distinct availability statistics, bidding such channels among SUs can be viewed as bidding multiple heterogenous objects. We first applied the UD auction and explored the instantaneous link condition of each SU over the primary channels for its throughput maximization. To avoid accessing primary channels with high load, we further proposed the LBUD auction, in which distributed learning of the primary channels at each SU is performed and incorporated in the auction mechanism. The proposed LBUD auction explores both channel availability statistics and instantaneous link gains of the SUs in order to maximize SUs' throughputs. It also reduces communication overhead of the required bidding data over the UD auction. Such a joint consideration of both primary channel availabilities and secondary link conditions is not considered in existing works. We showed that the

proposed LBUD auction is DSIC. We further proposed an adaptive price increment algorithm to improve convergence speed of the iterative procedure in the auction. Numerical results show the effectiveness of our proposed auction mechanism in terms of the throughput gain.

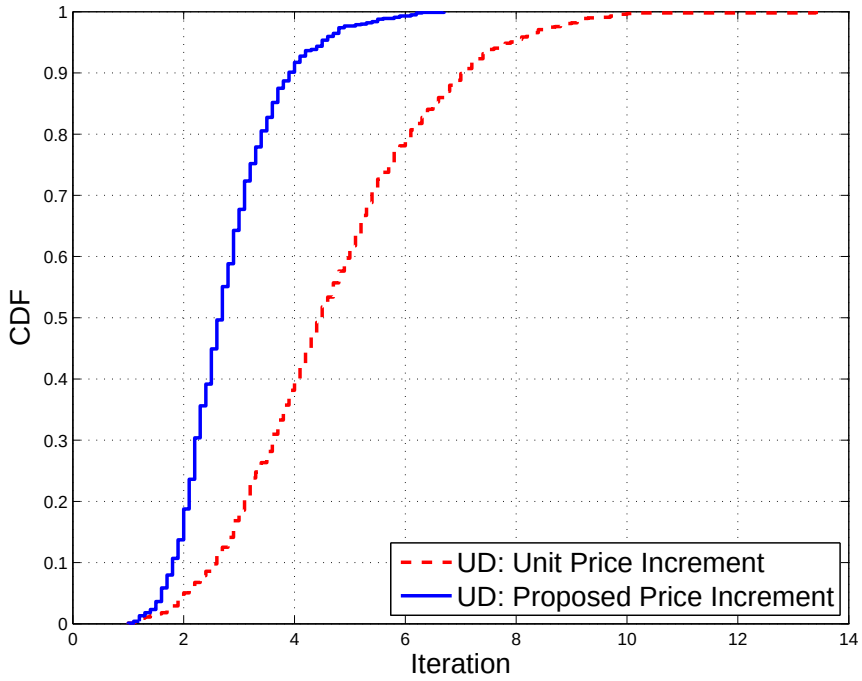

Figure 3.2: CDF of number of iterations under the UD auction ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 2, SNR $= 8$ dB), integer-valued case.

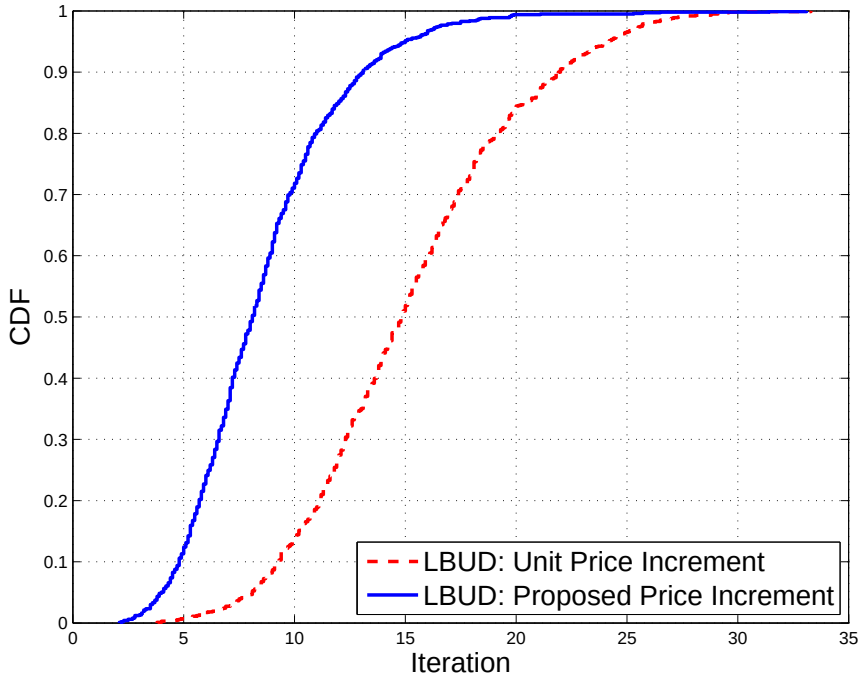

Figure 3.3: CDF of number of iterations under the LBUD auction ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 2, SNR $= 8$ dB), integer-valued case.

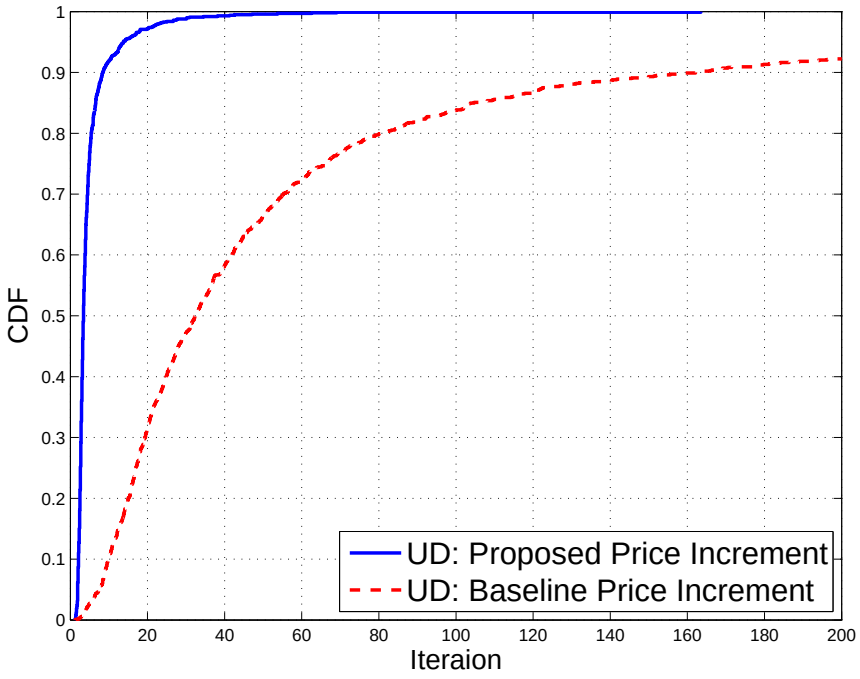

Figure 3.4: CDF of number of iterations under the UD auction ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 2, SNR = 8 dB), real-valued case.

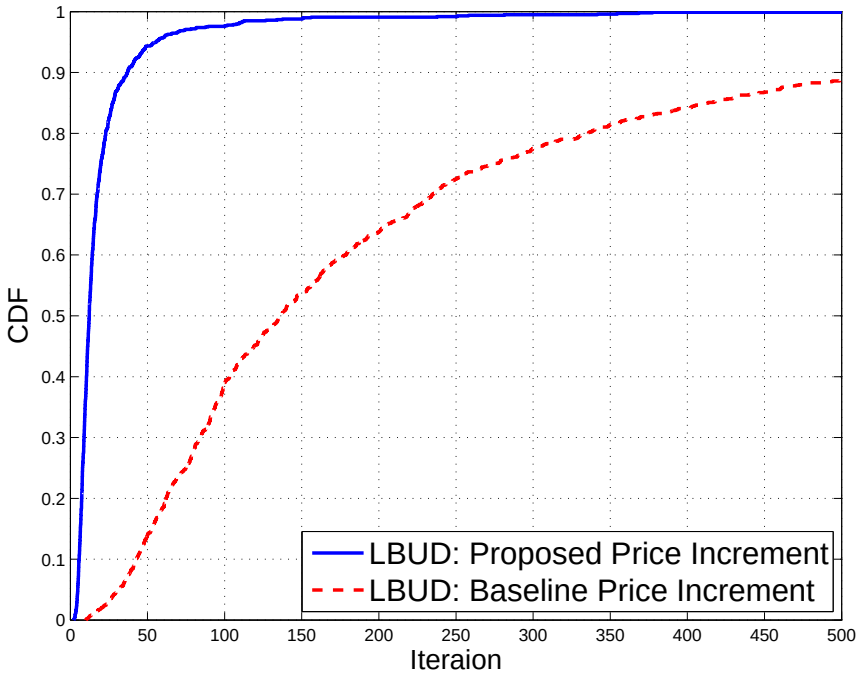

Figure 3.5: CDF of number of iterations under the LBUD auction ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 2, SNR = 8 dB), real-valued case.

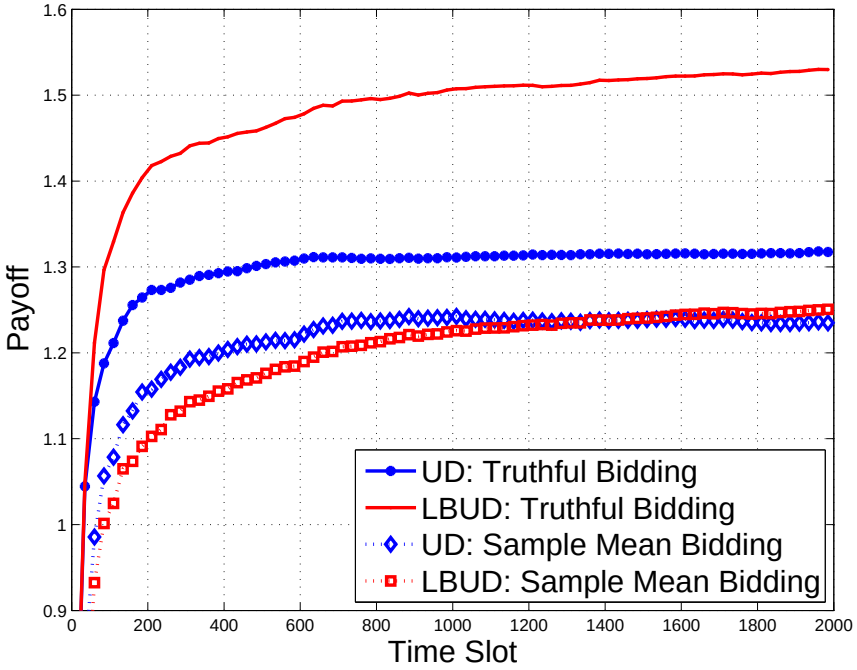

Figure 3.6: Average payoff per SU vs. time slot ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 3, SNR $= 8$ dB).

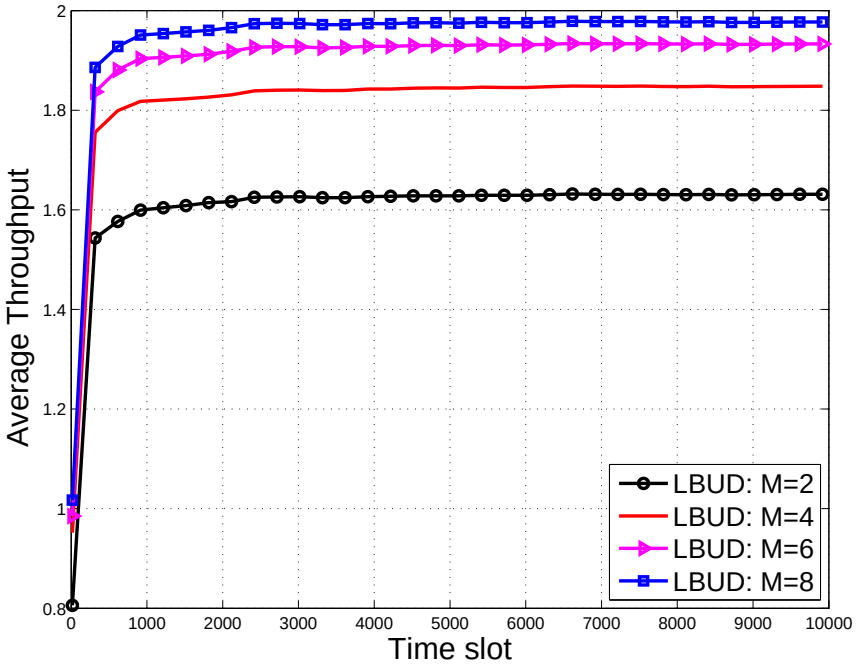

Figure 3.7: Average throughput vs. time slot ($N = 15$, $\boldsymbol{\theta}$: case 7, SNR $= 8$ dB).

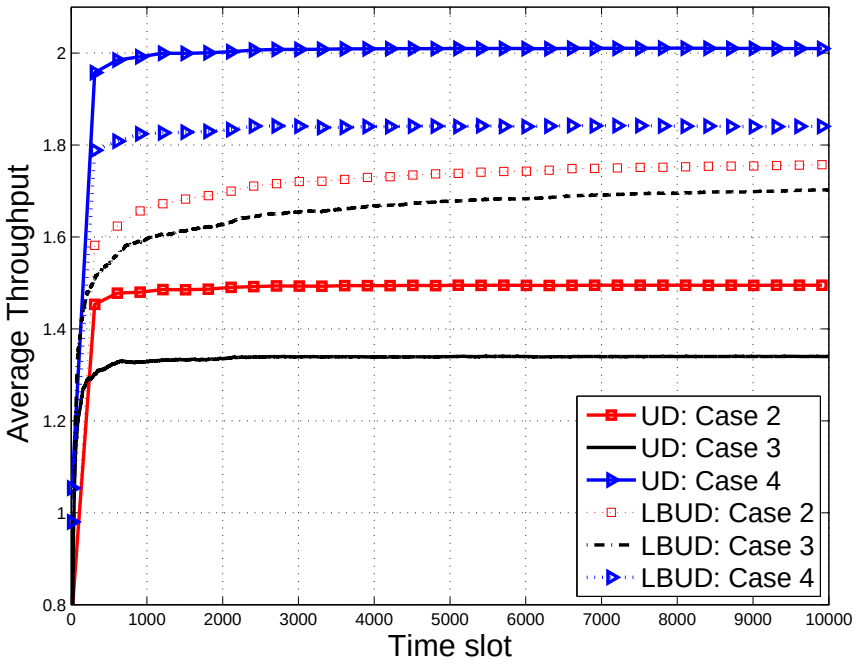

Figure 3.8: Average throughput vs. time slot ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 3, SNR = 8 dB).

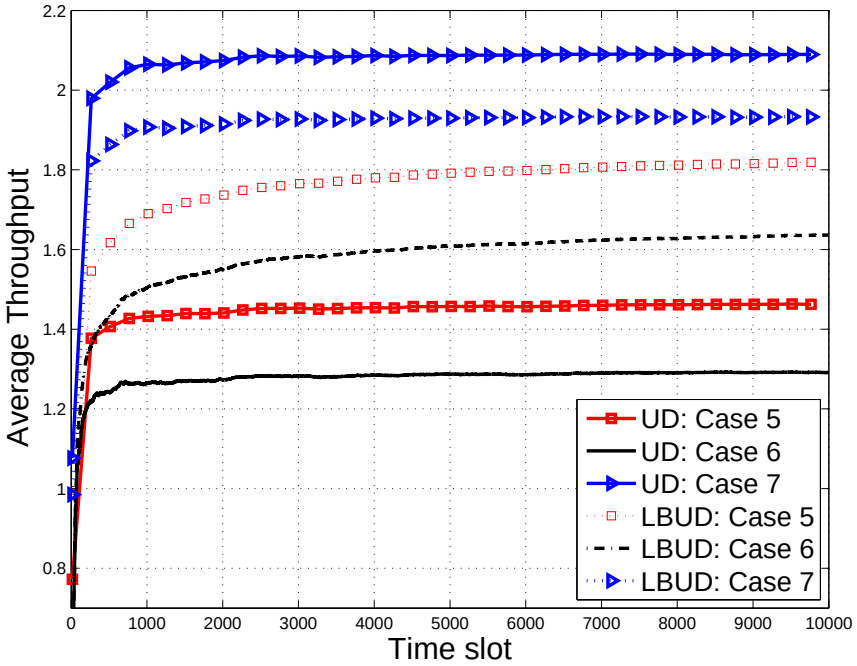

Figure 3.9: Average throughput vs. time slot ($N = 15$, $M = 6$, SNR = 8 dB).

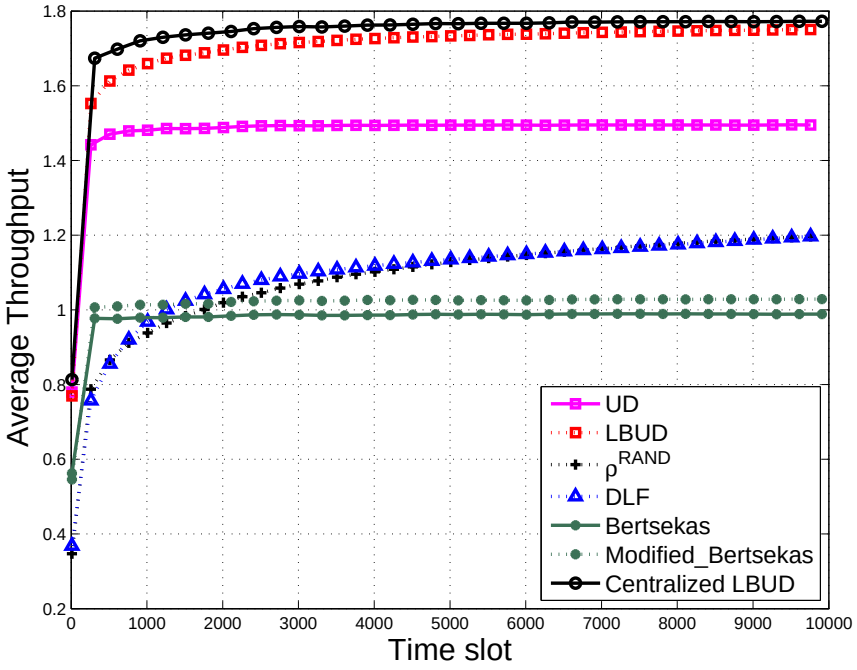

Figure 3.10: Average throughput vs. time slot ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 2, SNR = 8 dB).

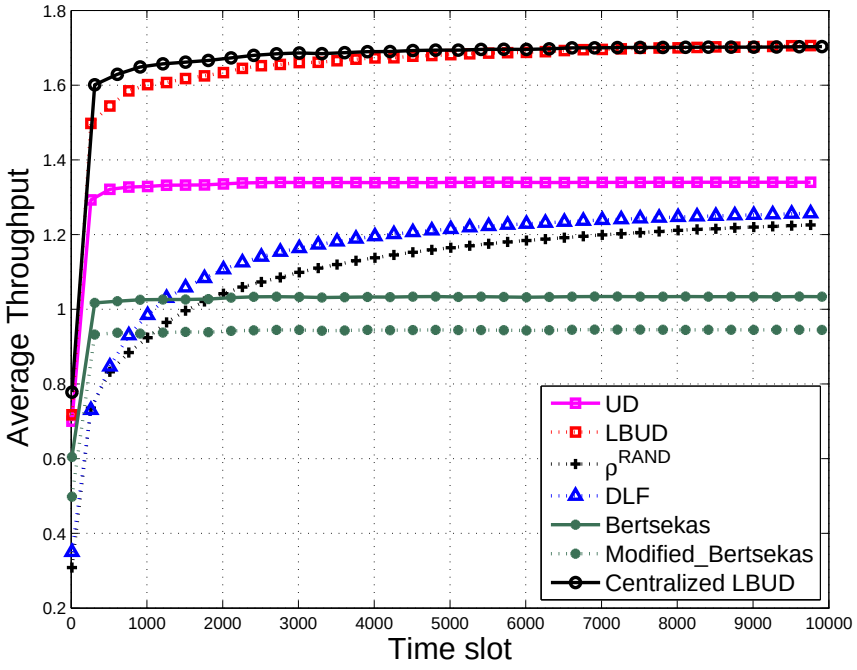

Figure 3.11: Average throughput vs. time slot ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 3, SNR = 8 dB).

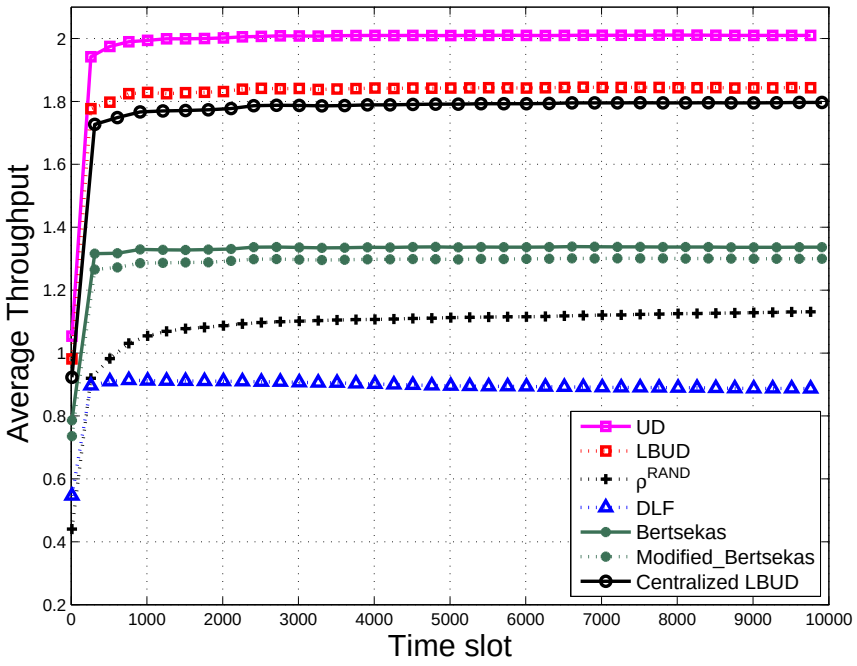

Figure 3.12: Average throughput vs. time slot ($N = 9$, $M = 4$, $\boldsymbol{\theta}$: case 4, SNR = 8 dB).

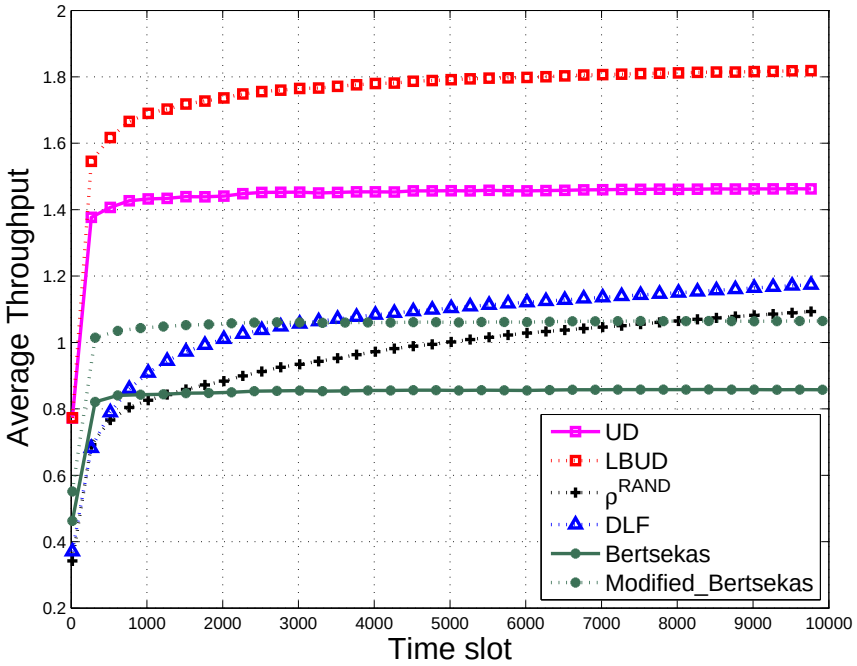

Figure 3.13: Average throughput vs. time slot ($N = 15$, $M = 6$, SNR = 8 dB, $\boldsymbol{\theta}$: case 5).

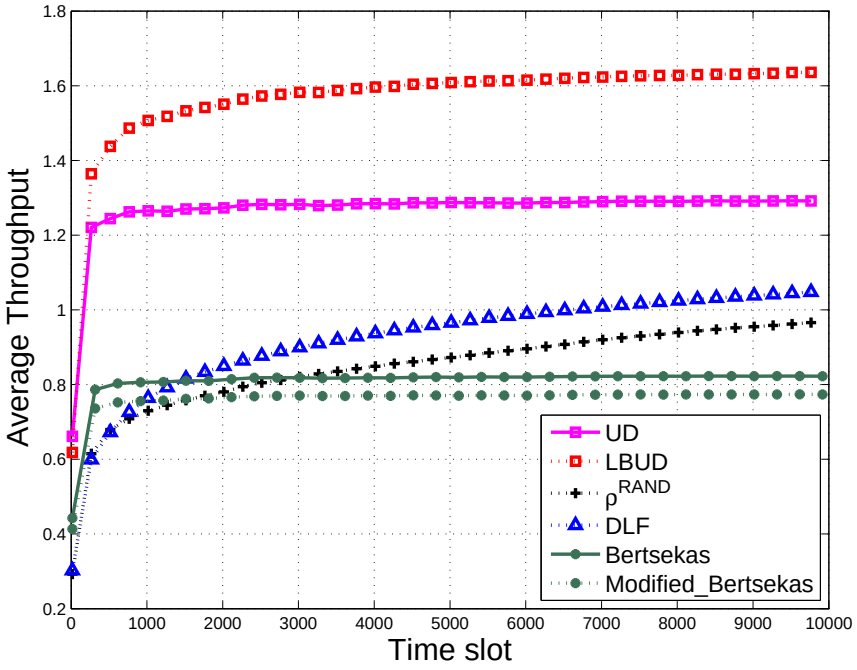

Figure 3.14: Average throughput vs. time slot ($N = 15$, $M = 6$, SNR = 8 dB, $\boldsymbol{\theta}$: case 6).

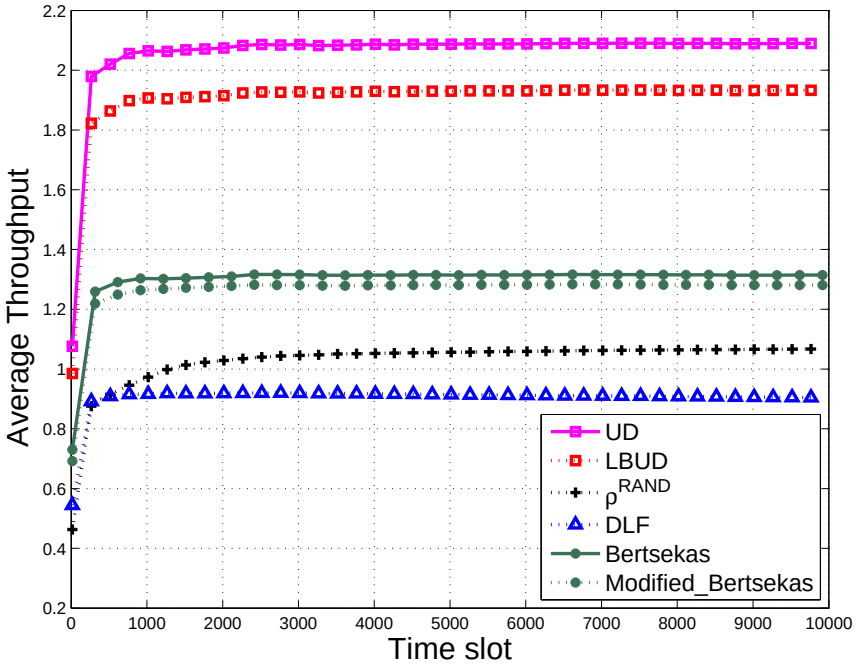

Figure 3.15: Average throughput vs. time slot ($N = 15$, $M = 6$, SNR = 8 dB, $\boldsymbol{\theta}$: case 7).

Chapter 4

Dynamic Spectrum Access via Multi-armed Bandit

4.1 Network Model

We consider a cognitive radio network where M decentralized cognitive secondary users compete over N orthogonal channels in a slotted primary network, where $N \geq M$. Let $X_i(n)$ be the availability state of the ith channel in the primary network at slot n, with

$$X_i(n) = \begin{cases} 1, \text{channel } i \text{ is available in slot } n \\ 0, \text{otherwise} \end{cases} \tag{4.1.1}$$

We assume $X_i(n)$ evolves over time as an i.i.d. Bernoulli random process with mean $\theta_i \in [0, 1]$. Let $\boldsymbol{\theta} \triangleq [\theta_1, \cdots, \theta_N]^T$. We assume $\boldsymbol{\theta}$ is unknown to all cognitive secondary users.

At time n, a secondary user (SU) j selects a channel i to sense and access if channel i is available. We assume perfect channel sensing. Let $T_i^j(n)$ denote the number of times that SU j senses channel i up to time n. SU j records $X_i^j(T_i^j(n)) = X_i(n)$ as its sensing observation of the availability of channel i. Then, vector $\mathbf{X}_i^j(n) \triangleq [X_i^j(1), ..., X_i^j(T_i^j(n))]^T$ hold the sensing observation history of SU j for channel i up to time slot n. Using these sensing observations, SU j obtains its estimation of θ_i at time slot n, denoted as $\hat{\theta}_i^j(T_i^j(n))$. Let $\hat{\boldsymbol{\theta}}^j(n) \triangleq [\hat{\theta}_1^j(T_1^j(n)), \cdots, \hat{\theta}_N^j(T_N^j(n))]^T$ be the vector of estimated mean of primary channels obtained at SU j.

For multiple SU's accessing the primary channels, we adopt a collision model in which an SU j's transmission is successful only if it is the sole user to access an available primary channel. In this case, we record a unit reward to SU j; otherwise, 0 reward is assigned.

The central problem for SU's dynamic accessing primary network is to design a decentralized access policy for SU's, based on each SU's local estimate of primary channel availabilities. The problem can be cast into a decentralized version of the classical MAB problem [39, 44]. The performance of an MAB policy is evaluated by a common measure called *regret*. It is defined as the difference between the total expected rewards of a genie-aided optimal decision (with $\boldsymbol{\theta}$ known) and that obtained by the any given policy

$$R(n, \boldsymbol{\theta}, M) \triangleq n \sum_{k=1}^{M} \theta_{(k)} - \sum_{i=1}^{N} \sum_{j=1}^{M} \theta_i E[S_i^j(n)] \tag{4.1.2}$$

where $\theta_{(k)}$ is the ordered version of θ_k, with $\theta_{(1)} >, \cdots, > \theta_{(N)}$, and $S_i^j(n)$ denotes the number of times, up to current slot n, that SU j is the sole user of channel i. Considering the above model, the design objective is to devise a decentralized policy that minimizes the regret, with no exchange of information among the SUs. For a

distributed access policy, each SU will select a channel to access based on its own estimate of the mean availability of channels.

In Section 4.2, we assume the total number of secondary users M is also unknown to all secondary users. Given the above model, a main problem is how to design decentralized policy that minimizes the regret, with no exchange of information among them. In Section 4.2, for a truly distributed access policy, each secondary user will select a channel to access based on its own estimate of $\hat{M}$ and the mean availability of channels $\hat{\theta}_i^j(T_i^j(n))$. In Section 4.3 and Section 4.4, it is assumed that the secondary user population is known to the SUs.

For decentralized dynamic spectrum access, there are two main problems for policy designs: sensing policy and access policy. The former provides the priority of primary channels to be selected for sensing at each SU, through learning of channel mean availability $\boldsymbol{\theta}$. Based on such priority, the latter determines the channel to be sensed and accessed through a collision-resolving mechanism. Thus, the sensing and access policies jointly affect the performance of decentralized dynamic access, and the choice of underlying sensing policy may affect the choice of access policy.

4.2 Distributed Opportunistic Spectrum Access with Unknown Population

4.2.1 Introduction

One of the main challenges in cognitive radio networks is to design dynamic spectrum access to efficiently utilize the spectrum. A hierarchical cognitive radio network

consists of primary users who are licensed to use the spectrum and the secondary users who opportunistically use the idle channels that are not occupied by the primary users. The channel availability statistics of the primary network are typically unknown to the secondary users. Through limited spectrum sensing, the secondary users search for idle channels and make decisions based on observation histories for channel access. In many scenarios, the secondary users are uncoordinated, ad hoc, and/or dissimilar. Thus, designing distributed policy for spectrum access among secondary users, where no information exchange or access arrangement among users, to maximize the total throughput of secondary users is critical. In this case, the challenges involved in dynamic spectrum access not only include online learning of the primary channel statistics based on local sensing observations, but also the distributed mechanism to resolve collisions among secondary users.

Consider a cognitive radio network with N independent channels and M secondary users, where $N \geq M$. For centralized scheduling of users' access, the problem of selecting the best M channels to maximize the throughput of secondary users under unknown channel availability statistics can be formulated as the classical Multi-armed Bandit (MAB) problem [39–41]. In this case, the problem is to design a policy to sequentially choose M plays of N arms with i.i.d. rewards over time. The performance of a MAB policy is evaluated by a metric called *regret*, defined as the difference in total expected rewards by the optimal choice and that by a given policy. For distributed access by the secondary users, the problem formulation can be viewed as the decentralized MAB problem. In contrast to the classic MAB problem, for decentralized MAB problem, M players compete over N arms. When multiple secondary users pick the same arm, collision occurs, thus resulting in lost rewards. To address this

problem, particularly arisen in dynamic spectrum access, a few decentralized learning and access policies were developed recently [9, 10, 20]. These policies use different mechanisms to achieve "coordination" among secondary users to orthogonalize their access to the M-best channels, and all achieve logarithmic growth of regret. Common to all these proposed distributed algorithms, the number of secondary users, M is assumed known to each secondary user. This information is utilized in determining the access decision to one of the M-best channels. Thus, although these algorithms are distributed in terms of learning of channel availability statistics based on local observation histories, secondary users share the common knowledge of the user population. In a practical dynamic environment, such knowledge may not be known and needs to be estimated and tracked at each secondary user in order to implement the distributed access policy.

In this chapter, we consider such a truly distributed spectrum access environment where both population of secondary users and channel availability statistics are unknown to secondary users, and population may change over time. We aim at developing a distributed mechanism for joint distributed access and user population estimation and tracking. Our particularly focus is on extending ρ^{RAND} policy proposed in [9] to the scenario with unknown user population and its online estimation. We first show that using distributed access policies with incorrect knowledge of M will result in linear growth of regret over time. In particular, the loss due to underestimation is more significant than that of overestimation, reflected in the rate of growth in regret. For distributed online learning of M, we propose a dynamic thresholding method in which collision counts are tested against thresholds in estimating M. The thresholds are dynamically adjusted using virtual systems built upon the current estimates of

mean channel availabilities. Our algorithm allows both overestimation and underestimation of M over time, and thus is capable of tracking the population change in a dynamic network environment.

Extensive research has been conducted in MAB problems. Both classical results for single play [39] and multi-play [40, 41] provide policies that are efficient in regret. Simple index-based policies were proposed [44, 62] which have logarithmic growth of regret. Motivated by dynamic spectrum access, decentralized policies among multiple players are proposed [9, 10, 20], all achieving logarithmic growth of regret. In [20], a TDFS policy is proposed to orthogonalize secondary users in a TDM fashion, which requires certain coordination among secondary users with the knowledge of M. In [9], a simple ρ^{RAND} policy is proposed to randomize each user's pick of the M-best channels. In [10], a DLF policy is proposed which is shown to have an order-optimal scaling with respect to M and N. However, the distributed collision-resolving mechanism relies on the knowledge of M and each secondary user's assigned id. Consequently, the DLF policy cannot be implemented in an unknown population environment. Closely related to our work, [9] also considered the scenario when M is unknown, and proposed ρ^{EST} policy based on ρ^{RAND} policy. The ρ^{EST} policy uses the idea of testing collision counts by a secondary user against thresholds to provide an estimate of M, but only at the conceptual level without any specific method given. A more detailed account of the differences between our work and [9] is given in Section 4.2.3.

4.2.2 Decentralized Spectrum Access Policies

The existing decentralized policies [9, 10, 20] can be viewed as variant distributed extensions of the UCB1 algorithm proposed in [44], which is a sample-mean based

index policy for the single user case. In the UCB1 algorithm, an index is assigned to every channel. The assigned index is a statistic which is based on the estimated sample mean of channel i and the total number of times that channel i has been visited up to the current time slot n. Let $T_i^j(n)$ denote the number of times that the secondary user j senses channel i up to time slot n. If the secondary user j picks channel i to sense at time slot n, then it obtains the value of $X_i(n)$ and records this value as $X_i^j(T_i^j(n))$. Let $\mathbf{X}_i^j(n) \triangleq [X_i^j(1), \cdots, X_i^j(T_i^j(n))]^T$ be the vector holding the sensing observations of secondary user j for channel i up to time slot n. With these sensing observations, secondary user j can estimate θ_i, the mean availability of channel i, at time n as in (3.4.1). Each secondary user j obtains an index called g-statistic for all the channels, $i = 1, \cdots, N$, as in (3.4.2). In the single user case ($M = 1$), this index will be used to rank all the channels and the user then selects the channel with the highest index at time n. When there are multiple secondary users, each user computes its own index vector of the channels $\mathbf{I}^j(n) \triangleq [I_1^j(n), \cdots, I_N^j(n)]^T$ based on its own observation history. Then, with certain distributed coordination designed differently by [9, 10, 20], each user will select a kth-best channel (*i.e.,* the channel with the kth highest ranking in $\mathbf{I}^j(n)$) to access, to ensure that the secondary users choose different channels but within the first M-best channels. Our proposed policy is closely related to ρ^{RAND} policy proposed in [9], which is explained below:

1. *Select channel to sense and access*: At every time slot n, each secondary user j obtain its ranking vector of primary channels $\mathbf{I}^j(n)$. It then selects the r_jth-best channel among the M-best channels to sense, where r_j is drawn from a uniform distribution: $r_j \overset{i.i.d.}{\sim} \text{Uniform}(M)$, for $j = 1 \cdots, M$. Let $\sigma(r_j, \mathbf{I}^j(n))$ be the channel index of the rth highest rank in $\mathbf{I}^j(n)$. If the channel is available

$X_{\sigma(r_j,\mathbf{I}^j(n))} = 1$, then the user can access the channel.

2. *Reselect channel under collision*: Each user j uses an acknowledgement for collision feedback. Let $\zeta_j(i,n) \in \{0,1\}$ denote such acknowledgment for user j on channel i, with $\zeta_j(i,n) = 1$ being collision among secondary users. Each user will redraw its rank $r_j \sim \text{Uniform}(M)$ only if there is collision in the previous transmission, otherwise, it will keep using the rank r_j generated previously.

4.2.3 Decentralized Spectrum Access with Unknown SU Population

When M is unknown, each secondary user j needs to obtain an estimate $\hat{M}_j$. It will then choose one of the $\hat{M}_j$ best channels to access. In the following, we will propose a dynamic thresholding policy π_{DT}, incorporating dynamic estimation of M and the distributed access in ρ^{RAND} under $\hat{M}_j$.

Linear Growth of Regret for $\hat{M}_j \neq M$

We first show that a fixed but inaccurate estimate $\hat{M}_j \neq M$ will lead to the linear growth of regret. This is in contrast to the logarithm growth under the perfect knowledge of M.

- *Underestimation*: Through online learning processes, the secondary users gather more information and obtain better estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$. It can be shown that, the probability of events that all users have the M-best channels asymptotically goes to 1. Among M-best channels, the users will try to orthogonalize the channel to access with each other. If only one user underestimate the user

population, *i.e.,* $\hat{M}_j < M$, for some j, there will be at least one collision among users every slot, provided all users share the same M-best channels. Thus, a smaller value of $\hat{M}_j$ will cause a linear growth of regret $O(n)$. The actual regret is lower bounded by this best scenario.

- *Overestimation*: Again, the probability of events that all users have the M-best channels asymptotically goes to 1. Overestimating M leads to the possibility of choosing the next $(\hat{M} - M)$-best channels, instead of M-best channels. This results in the difference in mean channel availabilities as compared to the optimal choice, which translates into linear growth of regret $O(n)$.

Even though both underestimation and overestimation lead to linear growth of regret, comparing the two, overestimation leads to lower regret than underestimation (*i.e.,* smaller leading coefficient of growth rate). To see this, for $\hat{M}_j < M$ case, collisions lead to zero reward (throughput); on the other hand, choosing from non M-best channels leads to an expected non-zero reward with the loss bounded by the difference of the mean availability of the $\hat{M}$th-best channel and the best channels. We will demonstrate in simulation the difference in performance loss in terms of regret between overestimation and underestimation cases.

Population Estimation through Thresholding

In a practical system, we need to dynamically estimate the user population, based on the observation history, to improve our knowledge of M. In [9], a ρ^{EST} algorithm is proposed under unknown M that is a modified policy of ρ^{RAND} policy. In ρ^{EST} algorithm, with the total transmission horizon T known, the estimate $\hat{M}_j$ for user j is updated by comparing the number of collisions so far to a threshold $\psi(T, \hat{M}_j)$. The

idea is that if $\psi(T, \hat{M}_j)$ is chosen properly, then for $\hat{M}_j < M$, the collision build-up will make the collision count exceed $\psi(T, \hat{M}_j)$, serving as a mechanism to increase $\hat{M}$. However, there are three aspects in ρ^{EST} that were not discussed:

- The policy and its property critically depends on the threshold $\psi(T, \hat{M}_j)$ setting. However, no explicit method for setting the thresholds was discussed; instead, $\psi(T, \hat{M}_j)$ was assumed given.

- The policy only allows upward increase of $\hat{M}_j$. Thus, to ensure accurate estimation, the threshold needs to be set large to avoid overestimation. This leads to slow learning. For finite horizon, since the user population will mostly be underestimated, it results in large regret as we have shown in the previous section.

- The policy cannot adapt to changing environment when M varies. This is due to the one-way upward increase mechanism of $\hat{M}_j$.

In this chapter, we try to address the above three issues in our policy design by proposing a method to set threshold, and dynamic updating $\hat{M}_j$ which allow users to track M in a changing environment.

Let n_o be the time slot of previous $\hat{M}_j$ update. Given the collision indicator $\zeta_j(i,t)$ at slot t on channel i, the total number of collisions experienced by secondary user j up to the current slot n can be obtained as

$$K_j(n; \hat{M}_j, \boldsymbol{\theta}) = \sum_{t=n_o+1}^{n} \sum_{k=1}^{\hat{M}_j} \zeta_j(\sigma(k, \mathbf{I}_j(t)), t). \tag{4.2.1}$$

Note that, besides being a function of $\hat{M}_j$, the total number of collisions $K_j(n; \hat{M}_j, \boldsymbol{\theta})$ is also a function of $\boldsymbol{\theta}$. Indeed, the distribution of entries in $\boldsymbol{\theta}$ will affect the accuracy

in (3.4.2) that ranks the channel availabilities based on their estimates $\hat{\theta}_i^j(T_i^j(n))$, $i = 1, \cdots, N$.

We will use a thresholding method, where the threshold reflects the expected collision counts when the the secondary user population is indeed the estimate $\hat{M}_j$. The collision counts $K_j(n; \hat{M}_j, \boldsymbol{\theta})$ will be tested against such threshold to determine whether to increase or decrease the current estimate $\hat{M}_j$.

Dynamic Thresholding Policy $\pi_{\mathbf{DT}}$

As mentioned, the threshold for each secondary user j should reflect the expected number of collisions under $\hat{M}_j$, *i.e.,* $\mathrm{E}[K_j(n; \hat{M}_j, \boldsymbol{\theta})]$. To determine $\mathrm{E}[K_j(n; \hat{M}_j, \boldsymbol{\theta})]$, we need the mean availability vector $\boldsymbol{\theta}$ which is unknown. To address this issue, we propose a dynamic thresholding method using virtual systems. In this case, instead of being fixed, the threshold is dynamic based on current sample mean channel availability $\hat{\boldsymbol{\theta}}_j$, and thus it is a function of n. We denote this dynamic threshold at slot n for user j as $\Delta_j(\hat{M}_j, n)$.

1. **Virtual System**

   For each secondary user j at slot n, we consider a set of virtual systems: $\{\mathcal{S}_{j1}(n), ..., \mathcal{S}_{jN}(n)\}$. Each virtual system $\mathcal{S}_{ju}(n)$ consists of N primary channels and u secondary users. Channel availability statistics for the N primary channels is denoted as $\boldsymbol{\theta}'_j = [\theta'_{j1}, \cdots, \theta'_{jN}]^T$, which is set to be equivalent to $\boldsymbol{\theta}'_j \equiv \hat{\boldsymbol{\theta}}_j(n)$, where

   $$\hat{\boldsymbol{\theta}}_j(n) \triangleq [\hat{\theta}^j_1(T^j_1(n)), \cdots, \hat{\theta}^j_N(T^j_N(n))]^T. \tag{4.2.2}$$

   In other words, the virtual system is build upon the primary network using the current estimate of channel availability in the original true system, and installs u secondary users in the system. For each virtual system $\mathcal{S}_{ju}(n)$, the secondary user population u is known to every user j', and each user uses the ρ^{RAND} policy for access, for $j' = 1, \cdots, u$. Similar to (4.2.1), the collision count from the beginning until the current slot n' is then given by

   $$K_{j'}(n'; u, \boldsymbol{\theta}'_j) = \sum_{t=1}^{n'} \sum_{k=1}^{u} \zeta_{j'}(\sigma(k, \mathbf{I}_{j'}(t)), t) \tag{4.2.3}$$

   for $j' = 1, \cdots, u$.

2. **Dynamic Thresholding** $\Delta_j(\hat{M}_j, n)$

   With the virtual system that is created using the current estimate $\hat{\boldsymbol{\theta}}_j(n)$ for each secondary user j, we can dynamically obtain the threshold $\Delta_j(\hat{M}_j, n)$. Running ρ^{RAND} policy on the virtual systems generated by secondary user j, the secondary users in the virtual system try to estimate their unknown $\boldsymbol{\theta'}_j$. After a given time horizon T', the average collision counts for the virtual systems can be obtained, which will be used to determine the threshold.

Denote $\bar{K}(n', u, \boldsymbol{\theta}'_j)$ as the average collision counts for the virtual system $\mathcal{S}_{ju}(n)$, averaged over all users, *i.e.,*

$$\bar{K}(n'; u, \boldsymbol{\theta}'_j) = \frac{\sum_{j'=1}^{u} K_{j'}(n'; u, \boldsymbol{\theta}'_j)}{u}. \tag{4.2.4}$$

It serves as an estimate of the expected number of collision in the system with u secondary users under ρ^{RAND} policy, *i.e.,* $\mathrm{E}[K_{j'}(n', u, \boldsymbol{\theta}'_j)]$. It has been shown in [9] that the expected number of collisions under ρ^{RAND} has logarithmic growth, *i.e.,* $\mathrm{E}[K_{j'}(n', u, \boldsymbol{\theta}'_j)] = O(\log n')$. Thus, we will set the dynamic threshold $\Delta_j(u, n)$ at slot n in the true system using this estimate as

$$\Delta_j(u, n) = \frac{\bar{K}(T'; u, \boldsymbol{\theta}'_j)}{\log T'}, \tag{4.2.5}$$

where T' is the time horizon used to run the virtual system.

To apply the virtual systems, at slot n, we apply the following steps

(a) In the original true system, a secondary user j obtains its estimate $\hat{M}_j$;

(b) User j will then activate two virtual systems, $\mathcal{S}_{j\hat{M}_j}(n)$ and $\mathcal{S}_{j(\hat{M}_j-1)}(n)$, and compute $\Delta_j(\hat{M}_j, n)$ and $\Delta_j(\hat{M}_j - 1, n)$ based on (4.2.4) and (4.3.5);

(c) User j will use $\Delta_j(\hat{M}_j, n)$ and $\Delta_j(\hat{M}_j - 1, n)$ to test against $K_j(n; \hat{M}_j, \boldsymbol{\theta})/\log n$ to determine whether to increase or decrease $\hat{M}_j$ by 1:
If $K_j(n; \hat{M}_j, \boldsymbol{\theta})/\log(n - n_o) > \Delta_j(\hat{M}_j, n)$, $\hat{M}_j = \hat{M}_j + 1$;
Else if $K_j(n; \hat{M}_j, \boldsymbol{\theta})/\log(n - n_o) < \Delta_j(\hat{M}_j - 1, n)$, $\hat{M}_j = \hat{M}_j - 1$;
Else no change.

If at slot n, secondary user j updates its estimate $\hat{M}_j$, the total collision count so far is discarded and secondary user j starts new collision counts afterward,

i.e., $n_o = n$ in (4.2.1). The details of the dynamic thresholding for population estimation and access policy is summarized in Algorithm 3.

From the above, we see that as $\hat{\boldsymbol{\theta}}_j(n) \overset{n}{\to} \boldsymbol{\theta}$, we have

$$\frac{\mathrm{E}[K_{j'}(T'; \hat{M}_j, \boldsymbol{\theta}'_j)]}{\log T'} \to \frac{\mathrm{E}[K_j(T' - n_0; \hat{M}_j, \boldsymbol{\theta}])}{\log(T' - n_0)}, \tag{4.2.6}$$

where the LHS expression is from the virtual system and the RHS expression is under the true system. Thus, the threshold in (4.3.5), for $u = \hat{M}_j$, provides an estimate of $\mathrm{E}[K_j(T' - n_0; \hat{M}_j, \boldsymbol{\theta}])/\log(T' - n_0)$ in the true system.

3. **Implementation Aspects**

From above, we see that we essentially obtain the threshold $\Delta_j(\hat{M}_j, n)$ based on the threshold generated from the proposed virtual systems. In the following, we discuss two parameters to be set in the actual system:

Time horizon T' The time horizon T' in virtual system can be either set with a predetermined fixed value or it can be changed dynamically. For setting T' as a fixed value, it needs to be large enough to guarantee that $\mathrm{E}[K_{j'}(T'; \hat{M}_j, \boldsymbol{\theta}'_j)]/\log T'$ has reached its steady value. This may require a large value of T', which will make the online population estimation process slow. On the other hand, we can dynamically determine T': Based on the variation of $\bar{K}(T'; \hat{M}_j, \boldsymbol{\theta}'_j)/\log T'$ in (4.3.5) over T', we can determine whether the current T' is sufficient for determine the current threshold $\Delta_j(\hat{M}_j, n)$.

Threshold $\Delta_j(\hat{M}_j, n)$ **updating frequency** In π_{DT} policy, every secondary user j needs to activate two virtual systems to obtain the current value of

$\Delta_j(\hat{M}_j, n)$ and $\Delta_j(\hat{M}_j - 1, n)$, which can be slow and also unnecessary. Again, we can do either periodic updating of $\Delta_j(\hat{M}_j, n)$, or dynamically determine when to update $\Delta_j(\hat{M}_j, n)$ based on the change of $\hat{\boldsymbol{\theta}}_j(n)$. For the latter, we note that if the value of $\hat{\boldsymbol{\theta}}_j(n)$ unchanged, the corresponding virtual systems will be the same. Thus, we can measure the change of $\hat{\boldsymbol{\theta}}_j(n)$ over time, and only update $\Delta_j(\hat{M}_j, n)$ when the change has deemed significant to trigger an update of $\Delta_j(\hat{M}_j, n)$.

"Heat-up" period for threshold comparison Due to few samples in the initial collision counting, the value $K_j(n; \hat{M}_j, \boldsymbol{\theta})/\log(n - n_o)$ since the last update of $\hat{M}_j$ may have a high variation, causing frequent update of $\hat{M}_j$. To avoid this, we set a "heat-up" window period W to collect initial collision counts, and only do threshold comparison when $n - n_o > W$. Within the window W, $\hat{M}_j$ will remain unchanged.

4. **A Network with Changing Population**

In all existing studies so far, the total number of the secondary users M is assumed fixed, either known or unknown. Since in practical applications there are scenarios that a secondary user can either join or leave the network, this assumption may not always hold. Therefore, there is a need for learning and access policies which are able to track the changes of M as well.

Our proposed π_{DT} policy can automatically track such population change in the network, as it allows both upward and downward change of $\hat{M}_j$, by testing the current collision counts against the dynamic threshold. In the simulation section, we will provide an example to demonstrate such tracking behavior.

4.2.4 Simulation Results

In this section, we present the simulation results obtained by using π_{DT} policy developed in this chapter. We assume a cognitive radio network with $N = 9$ channels and $M = 4$ secondary users. The channel availability $X_i(n)$ follows i.i.d. Bernoulli random process, for $i = 1, \cdots, N$.

To demonstrate the difference of overestimation and underestimation of M, we plot the two cases in Fig. 4.1. We set the mean channel availability vector $\boldsymbol{\theta} = [0.1, 0.2, \cdots, 0.9]^T$. We employ ρ^{RAND} policy, assuming a fixed estimate $\hat{M}_j$ as M, and plot the normalized regret over time for $\hat{M}_1 = \cdots = \hat{M}_4 > M$ and $\hat{M}_1 = \cdots = \hat{M}_4 < M$. As can be clearly seen, the difference in terms of regret between underestimation and overestimation is substantial when the estimate different from the true value only by 1, *i.e.,* $\hat{M}_j = M - 1$ and $\hat{M}_j = M + 1$.

In Fig. 4.2, we compare the obtained total regret, normalized against $\log n$, under π_{DT} with unknown M and ρ^{RAND} with known M. We also plot the regret using ρ^{RAND} with a fixed incorrect knowledge $\hat{M} = M-1$ or $\hat{M} = M+1$. We set the mean channel availability vector $\boldsymbol{\theta} = [0.1, 0.2, \cdots, 0.9]^T$. As can been seen, the regret under ρ^{RAND} policy has logarithmic growth, as shown in [9]. For unknown M, under π_{DT} policy with dynamic thresholding, we see that the regret grows super-logarithmic due to the impact of $\hat{M}$ estimation error over time. Comparing to the case with fixed $\hat{M} = M-1$ which has linear growth rate, we see a large improvement in regret under the proposed π_{DT} policy with dynamic thresholding for M estimation. The performance under the $\hat{M} = M + 1$ is better due to a fixed overestimation. However, since M is unknown, this fixed estimate cannot be ensured in reality, and the performance serves as a lower bound to our π_{DT} policy.

To see what the growth rate of regret under π_{DT}, in Fig. 4.3, we plot the total regret normalized as $\frac{R(n,\boldsymbol{\theta},M)}{\sqrt{n}\log n}$, which is shown approximately constant over the time horizon. This indicates that the growth rate of regret is approximately $O(\sqrt{n}\log n)$.

To show how the developed algorithm can track the change of secondary user population, we simulate a dynamic environment in which secondary users can join or leave the network. We assume $\boldsymbol{\theta} = [0.11, 0.12, \cdots, 0.19]^T$ and the number of secondary users M varies over time from $6 \to 2 \to 5$. Fig. 4.4 shows the $\hat{M}_j$ of two secondary users who stayed in the network during the entire time. As we see, our proposed algorithm is able to track the change of total number of the secondary users in the network.

4.2.5 Summary

In this chapter, we developed a truly distributed dynamic spectrum access mechanism under both unknown number of secondary users M and unknown mean channel availability $\boldsymbol{\theta}$. By designing thresholding mechanism for online estimation of M over time, we extend ρ^{RAND} policy [9] to the scenario with unknown user population. We show that using distributed access policies with fixed incorrect knowledge of M will result in linear growth of regret over time, with underestimation incurring more significant loss than overestimation does, reflected in the rate of growth in regret. The proposed thresholding method dynamically adjusts the threshold for M updates, using virtual systems built upon the current estimates of mean channel availabilities. Our algorithm allows both overestimation and underestimation in estimating M over time, and thus is capable of tracking the change of M, *i.e.,* population change.

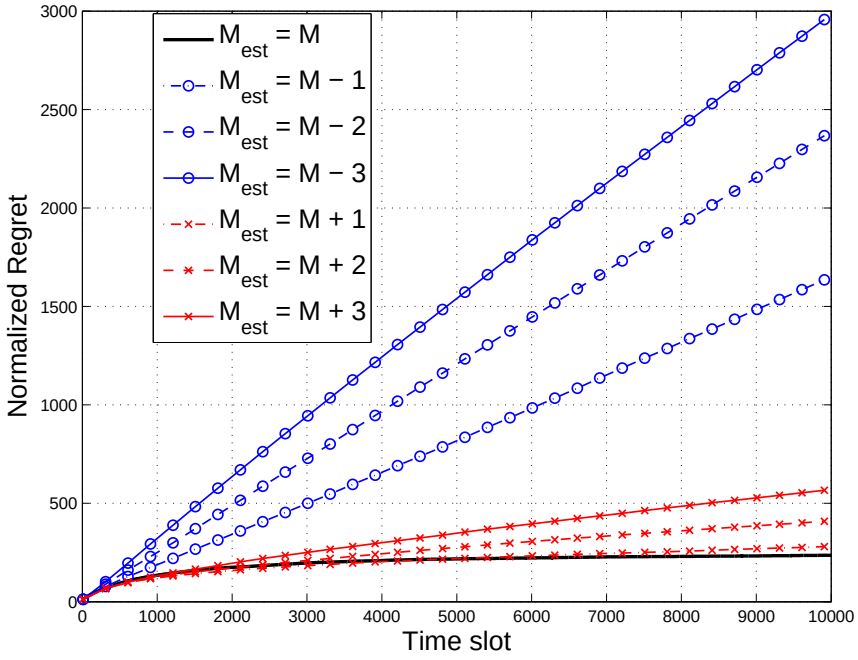

Figure 4.1: Normalized regret $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ for overestimation and underestimation of M.

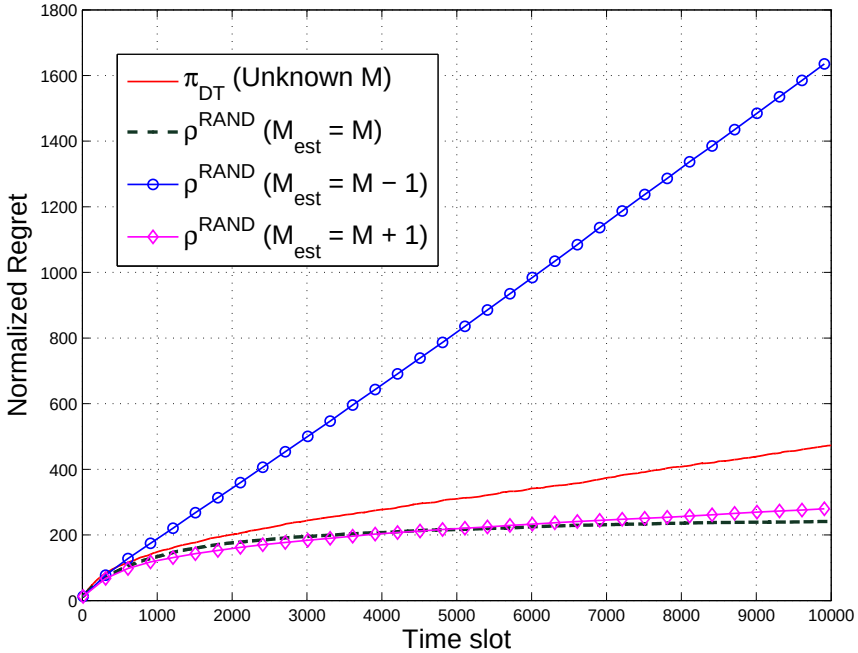

Figure 4.2: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ under ρ^{RAND} and π_{DT}. ($\boldsymbol{\theta} = [0.1, 0.2, ..., 0.9]$, M = 4, N = 9).

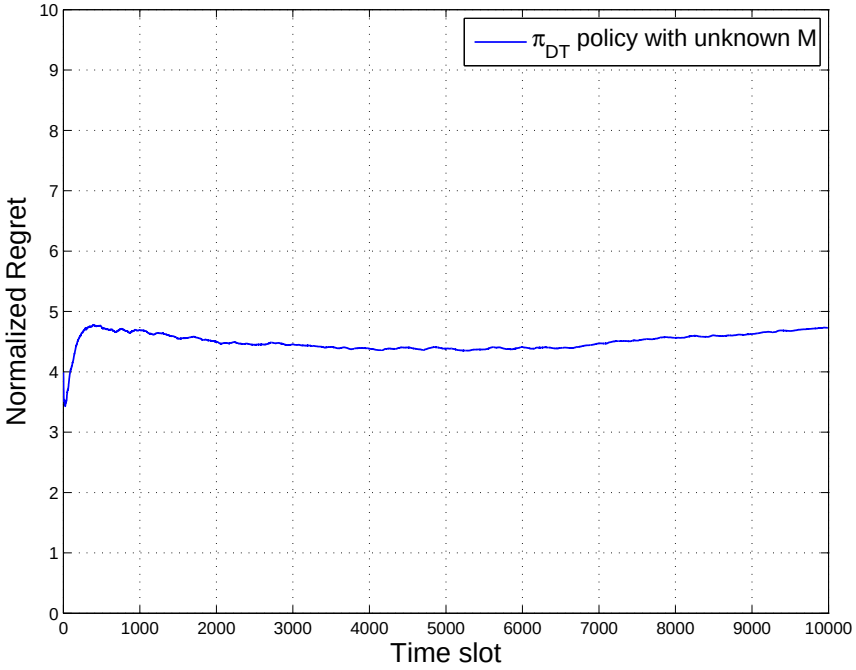

Figure 4.3: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\sqrt{n}\log n}$ under π_{DT} ($\boldsymbol{\theta} = [0.1, 0.2, ..., 0.9]$, M = 4, N = 9).

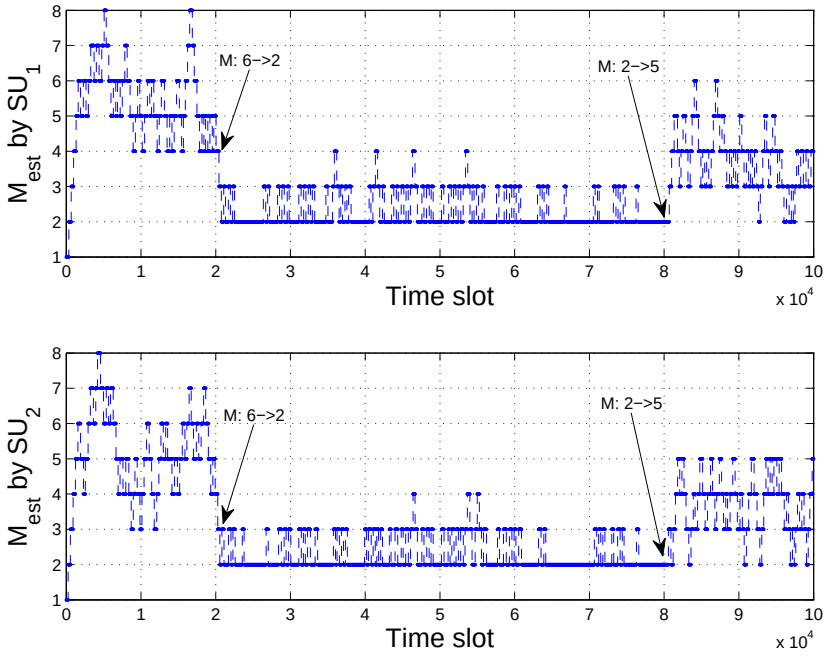

Figure 4.4: Trajectory of $\hat{M}_j$ for secondary users in the network in a dynamic network environment ($\boldsymbol{\theta} = [0.11, 0.12, \cdots, 0.19]^T$, $N = 9$)

Algorithm 3 Dynamic thresholding policy π_{DT} for each user j, under N channels and M secondary users.

1. **Input**: n: Current time slot
   T' Horizon length of the virtual systems
   n_o Time slot of the latest $\hat{M}_j$ update
   W: Soft window size
   $\hat{M}_j$: Current estimate of M by user j
   $\hat{\theta}_i^j(T_i^j(n))$: Sample mean availability of channel i for user j up to time slot n
   $I_j(i,n)$: g-statistic of channel i at time slot n for user j
   $\sigma(k, \mathbf{I}_j(n))$: Index of the k^{th} highest entry in $\mathbf{I}_j(n)$
   $\zeta_j(i,n)$:Collision indicator at time slot n on channel i

2. **Init**: Sense each channel once
   $n \leftarrow N+1$, $n_o \leftarrow N+1$, $\hat{M}_j \leftarrow 1$, $r_{\text{rank}} \leftarrow 1$, $i_{\text{chan}} \leftarrow N$, $\zeta_j(i,n) \leftarrow 0$ for all channels $i = 1, \cdots, N$.

3. **StartLoop** $n \leftarrow n+1$

4. **if** $\zeta_j(i_{\text{chan}}, n-1) = 1$ **then**
   Draw a new $r_{\text{rank}} \sim \text{Unif}(\hat{M}_j)$ **endif**
   Sense the channel and transmit if it is idle.
   Update $i_{\text{chan}} \leftarrow \sigma(r_{\text{rank}}, \mathbf{I}_j(n))$.
   if $\zeta_j(i_{\text{chan}}, n-1) = 0$ and $r_{\text{rank}} > \hat{M}_j$ **then**
   Draw a new $r_{\text{rank}} \sim \text{Unif}(\hat{M}_j)$ **endif**
   Sense the channel and transmit if it is idle.
   Update $i_{\text{chan}} \leftarrow \sigma(r_{\text{rank}}, \mathbf{I}_j(n))$

5. If collision, $\zeta_j(i_{\text{chan}}, n) \leftarrow 1$, otherwise, 0.

6. **if** $n - n_o > W$ **then**

   (a) Activate virtual systems $\mathcal{S}_{ju}(n)$, for $u \in \{\hat{M}_j - 1, \hat{M}_j\}$: Set $\boldsymbol{\theta}'_j \leftarrow \hat{\boldsymbol{\theta}}_j(n)$, N channels, run ρ^{RAND} policy for T' slots.

   (b) Update $\Delta_j(u,n)$, for $u \in \{\hat{M}_j - 1, \hat{M}_j\}$:
   $\bar{K}(n', u, \boldsymbol{\theta}'_j) = \frac{\sum_{j'=1}^{u} K'_j(n', u, \boldsymbol{\theta}'_j)}{u}$, $\Delta_j(u,n) = \frac{\bar{K}(T', u, \boldsymbol{\theta}'_j)}{\log T'}$

   (c) Obtain total collision counts from slot n_o up to n:
   $K_j(n; \hat{M}_j, \boldsymbol{\theta}) = \sum_{t=n_o+1}^{n} \sum_{k=1}^{\hat{M}_j} \zeta_j(\sigma(k, \mathbf{I}_j(t)), t)$.
   if $\frac{K_j(n;\hat{M}_j,\boldsymbol{\theta})}{\log(n-n_o)} > \Delta_j(\hat{M}_j, n)$ **then** $\hat{M}_j \leftarrow \hat{M}_j + 1$, $n_o \leftarrow n$
   endif
   if $\frac{K_j(n;\hat{M}_j,\boldsymbol{\theta})}{\log(n-n_o)} < \Delta_j(\hat{M}_j - 1, n)$ **then** $\hat{M}_j \leftarrow \hat{M}_j - 1$, $n_o \leftarrow n$
   endif

   endif

4.3 Learning-Stage Based Decentralized Adaptive Access Policy for Dynamic Spectrum Access

4.3.1 Introduction

Designing dynamic spectrum access to efficiently utilize the spectrum is one of the main objectives in cognitive radio networks. A hierarchical cognitive radio network consists of licensed primary users for accessing the spectrum and the secondary users who opportunistically use the idle channels that are not occupied by the primary users. Since the channel availability statistics of the primary network are typically unknown to the SUs, they rely on limited spectrum sensing to search for idle channels and make decisions based on observation histories for channel access. In designing a distributed policy for spectrum access among SUs, where there is no information exchange or access arrangement among users, the challenges involved not only include online learning of the primary channel statistics using local sensing observations, but also the distributed mechanism to resolve collisions among SUs.

Assume a cognitive radio network with N independent primary channels and M SUs, where $N \geq M$. For centralized scheduling of users' access, the problem can be formulated as the classical Multi-armed Bandit (MAB) problem [39–41]. The throughput loss over time due to learning of the primary channel statistics as compared to the ideal case with known channel statistics is measured by *regret*. The minimum growth of regret over time under an efficient learning algorithm is characterized in [40], and is shown to have a logarithm growth over time. For distributed access

by the SUs, the problem formulation can be viewed as the decentralized MAB problem. Motivated by dynamic spectrum access, decentralized policies among multiple players are proposed in [9, 10, 20]. These policies use different mechanisms to achieve "coordination" among SUs to orthogonalize their access to the M-best primary channels. They all achieve logarithmic growth of regret, which are order-optimal. Note that the efficiency of a learning algorithm is measured not only by the asymptotic growth rate of regret, but also by the scaling constant of the growth rate. All aforementioned decentralized policies are order-optimal with a logarithm growth rate of regret. However, they perform differently in terms of the scaling constant. Thus, further improvement should be with respect to the improvement on the scaling constant. In this work, we aim at improving the scaling constant of the growth rate by designing an access policy that is adaptive to different learning stages.

In this chapter, we design an adaptive decentralized access policy for spectrum access. In particular, we focus on modifying the ρ^{RAND} policy proposed in [9] which is a very simple distributed learning and access policy requiring least amount of coordination among users. By noticing that the learning accuracy of the primary channels affects the access collision statistics, we adapt the distributed access coordination among SUs at different stages of learning accuracy. Specifically, we exploit a "perceived population" by each SU to reduce collision events at different learning stages. We design a metric that measures the level of learning accuracy and use that as an indicator to adjust the "perceived population" by each SU. Simulations show that our proposed adaptive policy improves the scaling constant of the normalized regret and can provide substantial improvement over the ρ^{RAND} policy.

4.3.2 Decentralized Spectrum Access Policies

The UCB1 algorithm proposed in [44] is a sample-mean based index policy for the single user learning and access. The existing decentralized policies proposed in [9,10, 20] are considered as the extensions of the UCB1 algorithm to the distributed case. In UCB1 algorithm, channels are ranked at each SU using a statistic called g-statistic. Let $T_i^j(n)$ denote the number of times that the SU j senses channel i up to time slot n. If SU j selects channel i to sense at time slot n, then it obtains the value of $X_i(n)$ and records this value as $X_i^j(T_i^j(n))$. Let $\mathbf{X}_i^j(n) \triangleq [X_i^j(1), \cdots, X_i^j(T_i^j(n))]^T$ be the vector holding the sensing observation history of channel i up to time slot n at SU j. Using $\mathbf{X}_i^j(n)$, SU j estimates θ_i of channel i at time n as in (3.4.1). The g-statistic at SU j for channel i is defined as in (3.4.2). It will be used as an index for SU j to rank the channels. In the single user case ($M = 1$), using the above index, the user selects the channel with the highest index at time n. For multiple SUs, each user computes its own index vector $\mathbf{I}^j(n) \triangleq [I_1^j(n), \cdots, I_N^j(n)]^T$ based on its own observation history. Then, each user will select the channel with the k^{th} highest ranking in $\mathbf{I}^j(n)$ to access. Distributed learning policies [9, 10, 20] propose different mechanisms for access coordination to ensure that the SUs choose different channels but within the first M-highest indexed channels. Among these policies, the ρ^{RAND} policy in [9] is a very simple distributed policy, requiring the least amount of coordination among users and it is order-optimal. We aim to modify the ρ^{RAND} policy using adaptive learning to improve the performance. We briefly review the ρ^{RAND} policy below:

1. *Select channel to sense and access*: At slot n, each SU j obtains its ranking

vector $\mathbf{I}^j(n)$. It then selects the r_j^{th} best channel among the M-best channels to sense, where r_j is drawn from a uniform distribution: $r_j \overset{i.i.d.}{\sim} \text{Uniform}(M)$. Let $\sigma(r_j, \mathbf{I}^j(n))$ be the channel index of the r_j^{th} highest rank in $\mathbf{I}^j(n)$. If the channel is available, then SU j accesses the channel.

2. *Reselect channel under collision*: Each SU j uses an acknowledgement for collision feedback. SU j will redraw its rank $r_j \sim \text{Uniform}(M)$ only if there is collision in the previous transmission. Otherwise, it will keep using the rank r_j generated previously to determine which channel to access.

4.3.3 An Adaptive Learning Policy Based on Perceived Population

To measure the efficiency of a learning algorithm, we need to consider both the asymptotic growth rate of regret, denoted as $r(n)$, and the scaling constant of the growth rate, $\lim_{n\to\infty} R(n, \boldsymbol{\theta}, M)/r(n)$. It has been shown in [39] that an efficient learning algorithm for centralized MAB problems should have a logarithm growth rate of regret. All aforementioned decentralized policies are order-optimal with a logarithm growth rate of regret. What is unclear is how they perform in terms of the scaling constant. This differentiates the performance among these existing decentralized algorithms. Note that all the existing proposed policies rely on the exact knowledge of the secondary network population M to resolve collisions among SUs. We aim at improving the scaling constant of the growth rate by designing a learning policy that exploits a "perceived population" by each SU.

Define $U_j(n)$ to be a "perceived population" at SU j, *i.e.,* what the user perceives to be the current population. The user will use this parameter in determining the

primary channel to access. The "perceived population" is adaptive over time as a function of M: $U_j(n) = f(M, n)$. Note that using $U_j(n) \neq M$ for learning and access is equivalent to having an inaccurate estimate of M of the secondary network population, and in the long run, will lead to a linear growth rate of regret [21]. However, in the short time, it can improve the performance by reducing collision events. Fig. 4.5 shows an example of the impact of the population overestimation on the performance of the ρ^{RAND} policy, where $U_j(n) = M + 1$, $\forall j, n$. We see an improvement of the regret during the transient behavior at the early time slots.

To understand this behavior, we note that each SU learns the mean channel availabilities $\boldsymbol{\theta}$ over time for access decision in a decentralized fashion. There are two types of events contribute to the regret $R(n, \boldsymbol{\theta}, M)$: 1) *Not choosing M-best channels*: the channel i that SU j accesses has the mean availability θ_i that is not among the top M highest ones in $\boldsymbol{\theta}$; 2) *Collision among SUs*: distributed access results in collision and thus unsuccessful transmissions for all colliding users. Although the two types of events are correlated, the coordination mechanism in a decentralized access policy directly affects the type 2 event. Note that the SUs are more prone to collision at the beginning. This is because the estimate of the mean channel availability $\hat{\theta}_i^j(T_i^j(n))$ in (3.4.1) is very inaccurate, resulting in the channel ranking in $\mathbf{I}^j(n)$ varies over time. In other words, the k^{th} highest ranking in $\mathbf{I}^j(n)$ maps to different channel indexes more frequently. For SUs j_1 and j_2 selecting the channels which have the k_1^{th} and k_2^{th} ranks among M-highest values in $\mathbf{I}^j(n)$, they may collide in the next time slot, even though they do not collide in the current slot. At this stage, if we relax the constraint of selecting among the M-best channels to among the U-best channels, where $U > M$, it potentially decreases the collision among the SUs and also increases

the chance of selecting one of the true M-best channels. This is equivalent to use a larger "perceived population" $U_j(n) > M$ at SU j. As demonstrated in Fig. 4.5, by allowing larger perceived population, the regret improves at early time slots.

However, as the learning of $\boldsymbol{\theta}$ improves over time, the channel ranking in $\mathbf{I}^j(n)$ becomes more accurate and stable. Once two users select two different channels for access, they are likely to stay on the respective selected channels and remain collision free. In this case, using larger perceived population will increase the probability of selecting the channels outside of the M-best channels and hence has a negative impact on the throughput. Therefore, at this stage, it is necessary to use the true population as the "perceived population" for each SU.

Based on this analysis on the transient and long-term behavior, we propose an adaptive learning algorithms which adapt the "perceived population" $U_j(n)$ at each SU j to the different stages of learning of primary channel statistics to improve both the short-term and long-term regrets.

The main challenge in designing the adaptive learning algorithm is to determine the switching point for $U_j(n)$. We propose a thresholding method in determining $U_j(n)$. Let O_M be the set of indexes of the true M-best channels,

$$O_M = \left\{ i_m : \theta_{i_m} \in \{\theta_{(1)}, \cdots, \theta_{(M)}\},\ 1 \le m \le M \right\} \tag{4.3.1}$$

where $\{\theta_{(i)}\}$ is the ordered statistics of $\{\theta_i\}$ with $\theta_{(1)} > \cdots > \theta_{(N)}$. Let $\hat{O}_M^j(n)$ denote the set of indexes of the empirical M-best channels for SU j at time slot n,

$$\begin{aligned} \widehat{O}_M^j(n) = \Big\{ & i_m : \hat{\theta}_{i_m}^j(T_i^j(n)) \in \\ & \left\{ \hat{\theta}_{(1)}^j(T_i^j(n)), \cdots, \hat{\theta}_{(M)}^j(T_i^j(n)) \right\},\ 1 \le m \le M \Big\}. \end{aligned} \tag{4.3.2}$$

Now we denote $\delta_W^j(n)$ as the average number of estimated M-best channels in common

during a window period W for SU j given by

$$\delta_W^j(n) = \frac{\sum_{i=1}^{W} \left| \widehat{O}_M^j(n) \cap \widehat{O}_M^j(n-i) \right|}{W}, \quad n \geq W \tag{4.3.3}$$

where $0 \leq \delta_W^j(n) \leq M$. Denote the normalized version of $\delta_W^j(n)$ as $\bar{\delta}_W^j(n) = \delta_W^j(n)/M$. Denote $\Delta^j(n)$ as the cumulative moving average of $\bar{\delta}_W^j(n)$ as

$$\Delta^j(n) = \frac{\sum_{n'=W}^{n} \bar{\delta}_W^j(n')}{n - W} \tag{4.3.4}$$

where we have $0 \leq \Delta^j(n) \leq 1$. This quantity can be computed recursively based on the current $\bar{\delta}_W^j(n)$ and previous $\Delta^j(n-1)$ without the need to store all the history data as

$$\Delta^j(n) = \frac{1}{n-W}\bar{\delta}_W^j(n) + \frac{(n-W-1)}{n-W}\Delta^j(n-1), \quad \text{for } n \geq W. \tag{4.3.5}$$

As the estimate of $\boldsymbol{\theta}$ improves over time, the difference between $\widehat{O}_M^j(n)$ and O_M reduces. Thus, $\Delta^j(n)$ indicates the level of accuracy of the empirical M-best channels to the true M-best channels. In other words, the metric $\Delta^j(n)$ provides a measure of the learning accuracy over time.

The value of $\Delta^j(n)$ will be tested against thresholds $\{\tau_k\}$ to determine the switching point for $U_j(n)$, where $k = 1, \cdots, K$ and K indicates the total number of switching points used. We summarize the main steps of the modified ρ^{RAND} policy with adaptive learning and access, named Rand-ALC(K), below. Detailed description is shown in Algorithm 4.

1. Start with $U_j(n) = M + K$.

2. Compute $\Delta^j(n)$: At every time slot n, each SU j obtain $\Delta^j(n)$.

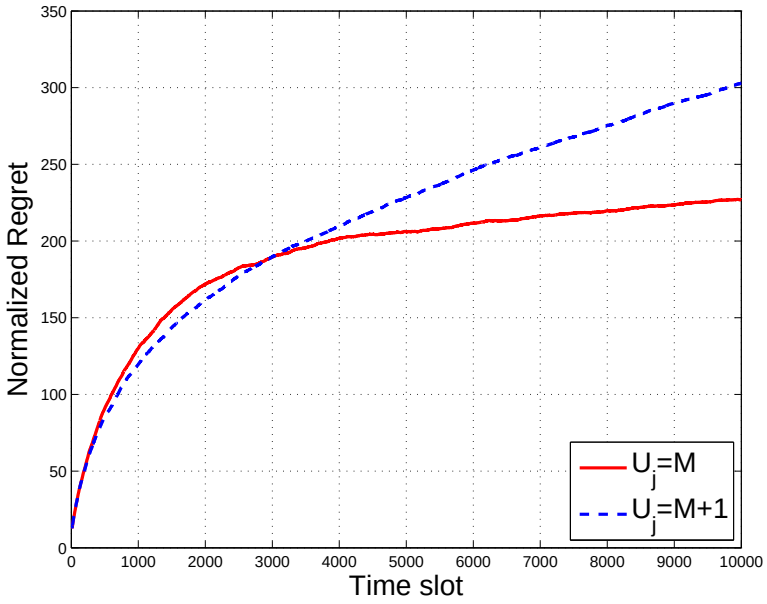

Figure 4.5: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ under the ρ^{RAND} policy using "perceived population" U_j, $\boldsymbol{\theta} = [0.1, 0.2, ..., 0.9]$, M = 4, N = 9.

3. Update $U_j(n)$: If $\frac{\Delta^j(n)}{M} \geq \tau_k$, then the SU j set $U_j(n) = U_j(n-1) - 1$, $k = k-1$, where τ_k is the current threshold used, and $1 \leq k \leq K$; otherwise, keep $U_j(n) = U_j(n-1)$.

4. Run the ρ^{RAND} policy (randomized access over the $U_j(n)$-best channel)

As we will see in simulations, using Rand-ALC(1) with $K = 1$ and single threshold τ is already effective in improving the throughput performance and regret.

4.3.4 Simulation Results

In this section, we present the simulation results obtained by the proposed policy. We assume a cognitive radio network with $N = 9$ channels and $M = 4$ SUs. The channel

Algorithm 4 Rand-ALC(K) policy for SU j

1. **Input**:
   n: Current time slot
   M: Number of SUs
   N: Number of channels
   T: Horizon length
   W: Window Size
   τ_k, $k = 1, \cdots, K$: Threshold
   $U_j(n)$: Perceived population of SU j at time slot n
   $\Delta^j(n)$: Level of learning accuracy of the empirical to the true M-best channels

2. **Init**: Sense each channel once
   $n \leftarrow N+1$, $U_j(n) \leftarrow M+K$, $\Delta^j(n) \leftarrow 0$, $k \leftarrow K$;

3. **Start Loop** $n \leftarrow n+1$

i) Update $U_j(n)$: If $\frac{\Delta^j(n)}{M} \geq \tau_k$, then $U_j(n) = U_j(n-1) - 1$, $k = k-1$; otherwise, $U_j(n) = U_j(n-1)$;

ii) Run ρ^{RAND} policy (randomized access over $U_j(n)$-best channel);

iii) Obtain $\widehat{O}_M^j(n)$: Set of indexes of empirical M-best channels for SU j at time slot n;

iv) Compute $\delta_W^j(n)$ as in (4.3.3), and compute $\bar{\delta}_W^j(n)$;

v) Update $\Delta^j(n)$ as in (4.3.5).
Stop Loop when $n = T$.

availability $X_i(n)$ follows i.i.d. Bernoulli random process, for $i = 1, \cdots, N$.

To demonstrate how the metric $\Delta^j(n)$ reflects the level of learning accuracy, in Fig. 4.6, we plot the trajectory of the averaged $\Delta^j(n)$ over time. We set the mean channel availability randomly as $\boldsymbol{\theta} = [0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7]^T$, and window size $W = 10$. We fix $U_j(n)$ value over time, and let each user implements the ρ^{RAND} policy with the "perceived population" $U_j(n)$, where $U_j(n) = M$, $M+1$, or $M+2$. As we see, the trajectory of averaged $\Delta^j(n)$ shows two stages of learning

at different rates, the initial learning with much faster rate of improvement, and then switched to a slower learning speed. In addition, we see that the rate of learning is not sensitive to the variation of $U_j(n)$.

In Fig. 4.8, we compare the normalized regret $R(n, \boldsymbol{\theta}, M)/\log n$ under the proposed Rand-ALC(1) policy and the ρ^{RAND} policy. We also compare them with Rand-ALC$^{\text{gen}}$(1), a genie-aided policy where we use normalized regret curve under fixed $U_j(n) = M$ and $U_j(n) = M + 1$ (e.g. in Fig. 4.5) to find the switching time n_{sw} for $U_j(n)$ from $M + 1$ to M to produce a lower regret. The same $\boldsymbol{\theta}$ value as in Fig. 4.6 is used in Fig. 4.8. We see that, our proposed policy with threshold $\tau = 0.98$ substantially outperforms the ρ^{RAND} policy in both transient and long-term behavior. Over 30% improvement is seen in long-term normalized regret, which indicates the improved scaling constant of the growth of regret. The performance of our proposed policy also tracks that of the genie-aided Rand-ALC$^{\text{gen}}$(1) policy very closely.

Similar to the experiment above, Fig. 4.9 shows the normalized regret under a different mean channel availability statistics, where $\boldsymbol{\theta} = [0.51, 0.52, \cdots, 0.59]^T$, *i.e.,* very similar mean statistics among the channels. As can be seen, our proposed policy again substantially outperforms the ρ^{RAND} policy (20% improvement) and approaches the genie-aided Rand-ALC$^{\text{gen}}$(1) policy.

4.3.5 Summary

We consider the problem of decentralized online learning and channel access in a cognitive radio network. Based on the existing distributed access policy, the ρ^{RAND} policy, we propose an adaptive decentralized access policy Rand-ALC(K). It adjusts the distributed coordination mechanism among SUs by adaptively changing the "perceived

population" at each SU to reduce collisions at different learning accuracy stages. We design a metric that measures the level of learning accuracy and use that as an indicator to adjust the "perceived population" by each SU. Simulations show that our proposed adaptive policy improves the scaling constant of the normalized regret and can provide substantial improvement over the ρ^{RAND} policy.

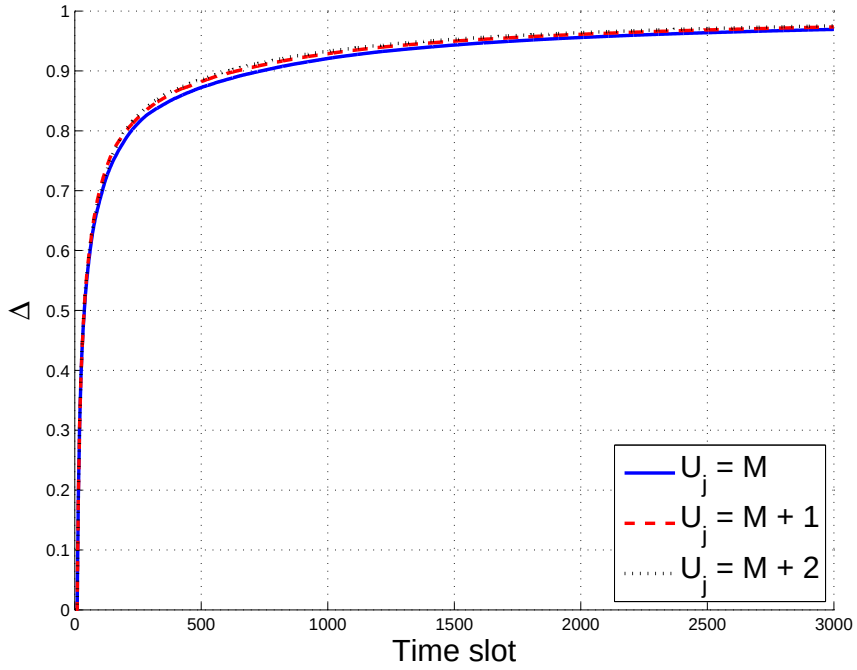

Figure 4.6: Average $\Delta^j(n)$ vs. time slot n. ($W = 10$, $\boldsymbol{\theta} = [0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7]$, M = 4, N = 9)

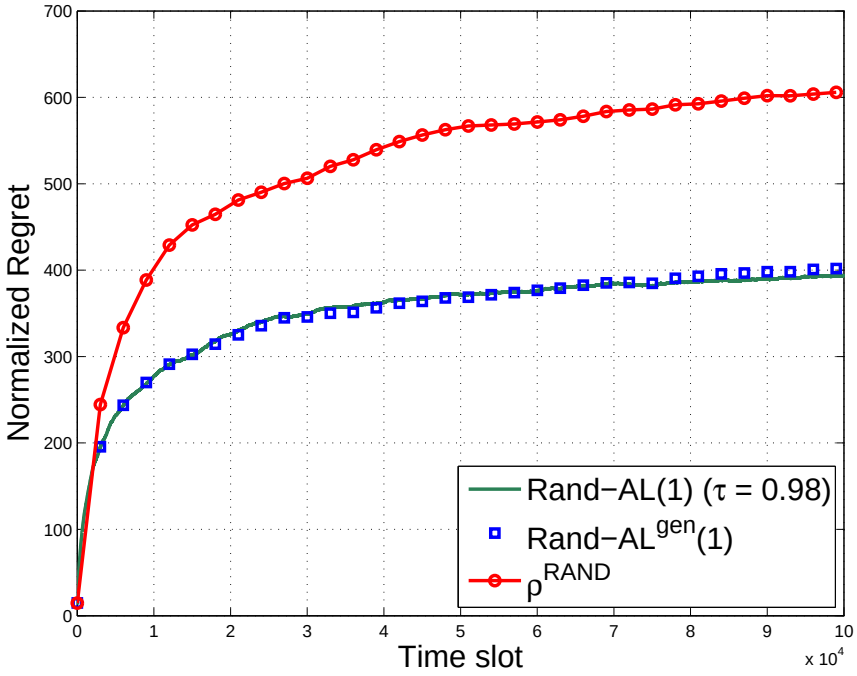

Figure 4.7: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ vs. time slot n. ($\boldsymbol{\theta} = [0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7]$, M = 4, N = 9)

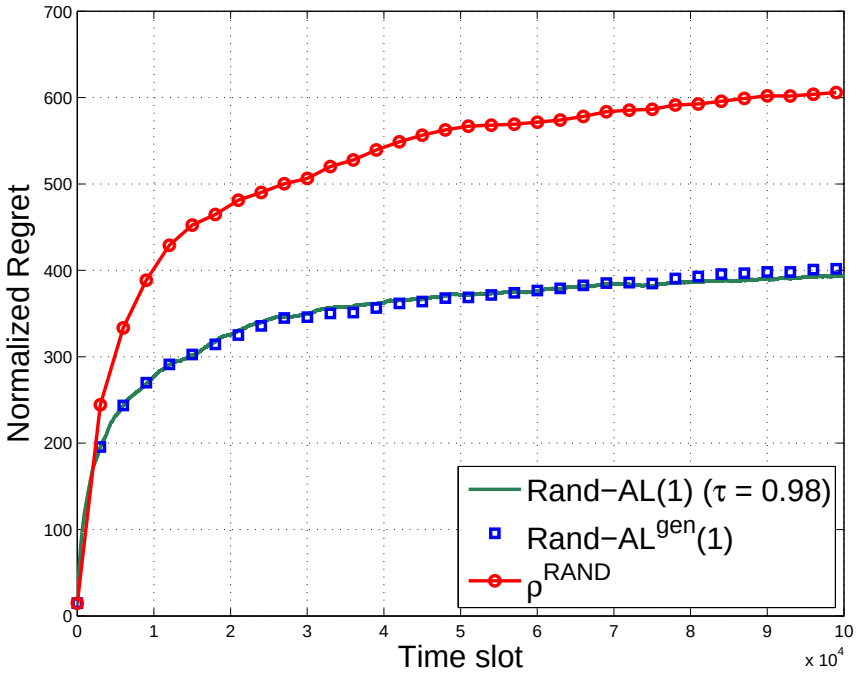

Figure 4.8: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ vs. time slot n. ($\boldsymbol{\theta} = [0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7]$, M = 4, N = 9)

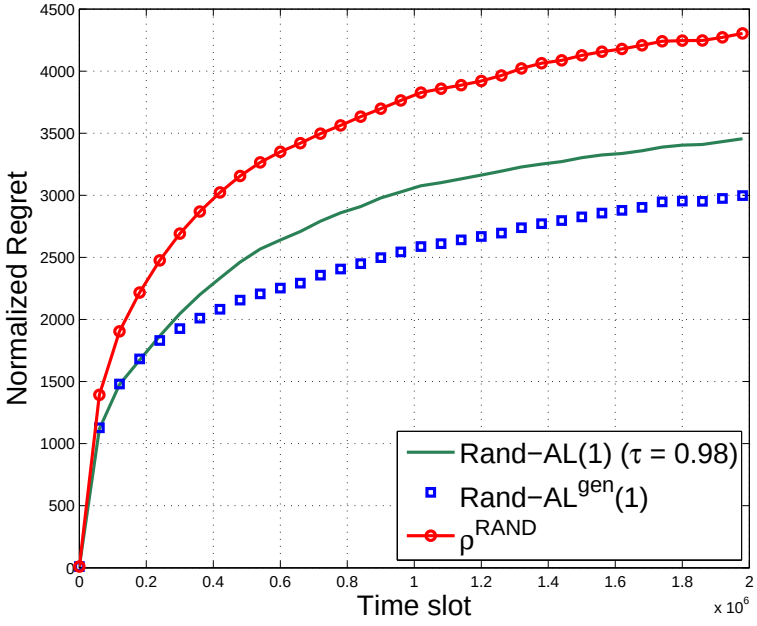

Figure 4.9: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ vs. time slot n ($\boldsymbol{\theta} = [0.51, 0.52, ..., 0.59]$, M = 4, N = 9).

4.4 Decentralized Spectrum Learning and Access Adaptive to Channel Availability Distribution in Primary Network

4.4.1 Introduction

Consider a cognitive radio network with N independent primary channels and M secondary users, with $N \geq M$. One of the main challenges in a such network is to provide efficient distributed dynamic spectrum access to utilize the available spectrum. Since the primary channel availabilities are typically unknown to the secondary users, they rely on limited spectrum sensing to search for idle channels and make decisions based on observation histories for channel access. Thus, the challenges in designing a distributed policy for spectrum access among secondary users involve not only online learning of the primary channel statistics using local sensing observations, but also the distributed mechanism to resolve collisions among secondary users.

For centralized scheduling of secondary users' access, classical Multi-Armed Bandit (MAB) [39–41] has been used to formulate the problem. The policy design is to find sequential decisions of M plays of N arms with i.i.d. rewards over time. In contrast, the distributed access among the secondary users can be viewed as the decentralized MAB problem, where M players compete over N arms. When multiple players choose the same arm, a collision occurs, thus resulting in lost rewards. To address this problem in distributed dynamic spectrum access, several decentralized learning and access policies have been recently developed [9, 10, 20]. These policies use different mechanisms to "resolve" collision among secondary users for their access to the M

most available primary channels. Although all these policies are shown to be order-optimal in the sense that they achieve the logarithmic growth of regret (a measure of the difference between the total expected reward of the genie aided optimal decision and the expected reward obtained by a policy), their relative performance is different due to different leading constants in the growth of regret. Although there are some analysis of these policies on how the growth of regret changes with M and/or N, there is no existing research on how the distribution of mean availability of primary channels affects the performance of these policies. Indeed, as will be shown in our study, among these existing policies, a policy may be more effective than the other for certain type of mean availability distribution of the primary channels, but not so for other distributions. In practice, it is desirable to design learning and access policies that perform well for a wide range of primary channel mean availability distributions. In this work, we aim at analyzing this effect and develop a learning and access policy that can be effective in various distributions of the mean channel availabilities.

In this chapter, we consider the effect of the mean availability distribution of primary channels on the performance of distributed learning and access policies. Our analysis focuses on two existing distributed access policies, namely the ρ^{RAND} policy [9] and the distributed learning with fairness (DLF) policy [10]. We first extend the recently proposed Bayesian learning automata (BLA) algorithm [42] to distributed online learning of underlying primary channel availabilities, and modify existing access policies to form BLA-ρ^{RAND} and BLA-DLF policies. We analyze the difference in the distributed access collision mechanism offered by the ρ^{RAND} and DLF policies. They can be considered as either passive or active correcting mechanism that is effective in certain type of mean channel availability distribution. Based on this, we propose a

dynamic switching learning and access (DSLA) policy that adapts to different channel availability distribution condition. Based on a closeness factor we propose, the DSLA policy automatically switches between the underlying learning algorithms, *i.e.,* upper confidence bound (UCB) [44] and BLA, as well as the access policies, ρ^{RAND} and DLF, to determine which policy is most effective for a given primary channel condition. Simulation studies show that our proposed DSLA policy is effective and provides good performance for a wide range of primary channel availability distributions.

4.4.2 Underlying Learning Policies: UCB vs. BLA

Upper Confidence Bound-1 (UCB1) Policy

Note that a sensing policy is essentially a learning algorithm of $\boldsymbol{\theta}$ that provides trade-off of exploration and exploitation based on history of observations. A UCB1 algorithm proposed by Auer *et al.* [44] is a single-user index-based policy. At time slot n, an index, which is a statistic called g-statistic, is assigned to each primary channel. It is defined for channel i as $I_i(n) \stackrel{\Delta}{=} \hat{\theta}_i(T_i(n)) + \sqrt{\frac{2\log n}{T_i(n)}}$, where $T_i(n)$ and $\hat{\theta}_i(T_i(n))$ are defined the same as $T_i^j(n)$ and $\hat{\theta}_i^j(T_i(n))$, except $j = 1$ in the single SU case. The SU then selects the channel with the highest g-statistic. Note that, the UCB1 is an order-optimal algorithm in the sense that the algorithm is shown to achieve the logarithmic growth of the regret [39, 44].

The UCB1 algorithm for single-user MAB with N-arms has recently been adapted to the decentralized MAB (DMAB) formulation in dynamic spectrum access [9,10,20] as the underlaying sensing policy for each distributed SU, as shown in Section 4.4.3

BLA Policy

A BLA algorithm was recently proposed for a classic MAB problem with $N = 2$ and each arm (*e.g.*, channel) with Bernoulli distribution [42]. The BLA algorithm is constructed based on the Bayesian inference with highly computationally efficient updating rules. The updating rules rely on updating the hyperparameters of the conjugate prior distributions for $X_i(n)$. Specifically, a beta distribution with two positive parameters (α, β), is assigned to each arm. The PDF of the corresponding beta distribution for arm i is given as

$$f_i(z) = \frac{z^{\alpha_i - 1}(1-z)^{\beta_i - 1}}{\int_0^1 y^{\alpha_i - 1}(1-y)^{\beta_i - 1} dy}, z \in [0,1];\ i = 1, 2. \tag{4.4.1}$$

At each arm i, a random variable z_i is drawn independently from $f_i(z; \alpha_i, \beta_i)$. The arm i with the highest value of z_i is selected, *i.e.,* $i^* = \arg\max_i \{z_i\}$. If a reward (e.g., bits successfully transmitted) is obtained from arm i^*, then α_{i^*} is increased by 1, *i.e.,* $\alpha_{i^*} = \alpha_{i^*} + 1$; otherwise $\beta_{i^*} = \beta_{i^*} + 1$.

It is shown in [42] that the BLA algorithm also achieves logarithmic growth of the regret. In addition, it outperforms the UCB-Tuned algorithm (a better learning algorithm than UCB1) [44] for all settings of the primary channel availabilities except the case that the two channels have similar availabilities combined with the high variance.

4.4.3 Distributed Access Policies

UCB-based Distributed Access Policies

Two recently proposed distributed access policies, ρ^{RAND} policy [9] and DLF policy [10], extend the single-user UCB1 policy to distributed ones for decentralized dynamic

spectrum access. In this case, each SU j obtains its sample mean channel availability as in (3.4.1) and its own g-statistic for channel i as in (3.4.2). Denote $\mathbf{I}^j(n) \triangleq [I_1^j(n), \cdots, I_N^j(n)]^T$. It is used by SU j to make its own ranking of the primary channels. To avoid collision among SU's due to selecting the same channel, the access policy is designed to provide a mechanism to "coordinate" SU's selection of channels. The mechanism essentially ensures that each SU selects a different channel among its M highest ranked channels:

- UCB-ρ^{RAND} policy [9]: Each SU j picks a random value r_j uniformly from 1 to M. It will then access the channel i which ranks the r_j^{th}-highest in $\mathbf{I}^j(n)$. At time slot n, if a collision occurred in the previous slot, SU j will re-draw $r_j \overset{i.i.d.}{\sim} \text{Uniform}(M)$; Otherwise, it keeps the previously generated rank r_j.

- UCB-DLF policy [10]: At time slot n, SU j selects the r_j^{th}-rank channel to access among the top M ranked channels in terms of $\mathbf{I}^j(n)$, where $r_j = ((j+n) \mod M) + 1$.

BLA-Based Distributed Access Policies

The existing access policies are all based on UCB1 policy for the underlaying sensing algorithm. In this chapter, we first extend the single-user BLA algorithm to the distributed multiple SU scenario with arbitrary N, and adapt ρ^{RAND} and DLF policies to be built upon the BLA algorithm, namely BLA-ρ^{RAND} and BLA-DLF, respectively.

Specifically, at time slot n, each SU j assigns a beta distribution with parameters $\alpha_i^j(T_i^j(n)), \beta_i^j(T_i^j(n))$ to each channel i, whose PDF is given by

$$f_i^j(z; T_i^j(n)) = \frac{z^{\alpha_i^j(T_i^j(n))-1}(1-z)^{\beta_i^j(T_i^j(n))-1}}{\int_0^1 y^{\alpha_i^j(T_i^j(n))-1}(1-y)^{\beta_i^j(T_i^j(n))-1}dy} \tag{4.4.2}$$

for $z \in [0,1]$. SU j generates a sample realization $z_i^j(n)$ from $f_i^j(z; T_i^j(n))$ for channel i, $\forall i$. Let $\mathbf{z}^j(n) = [z_1^j(n), \cdots, z_N^j(n)]^T$. It is used by SU j to rank the primary channel for sensing and access.

Two modified distributed access policies based on the above distributed BLA algorithm are described below.

1. **BLA-DLF Policy**

   With the underlying sensing policy based on the BLA algorithm, we modify the DLF policy to the BLA-DLF policy as follow.

   - *Select channel to sense and access*: At time slot n, SU j selects the r_j^{th}-rank channel among the top M channels ranked using $\mathbf{z}^j(n)$, where $r_j = ((j+n) \mod M) + 1$. The corresponding channel index $i^* = \{i : z_i^j(n) \text{ is } r_j^{th} \text{ highest in } \mathbf{z}^j(n)\}$. If $X_{i^*} = 1$, then the SU j will access the channel.

   - *Update the hyper parameters*: The parameters of PDF in (4.4.2) are updated as follows

   $$\begin{cases} \alpha_{i^*}^j(T_{i^*}^j(n)) = \alpha_{i^*}^j(T_{i^*}^j(n-1)) + 1, & \text{for } X_{i^*} = 1 \\ \beta_{i^*}^j(T_{i^*}^j(n)) = \beta_{i^*}^j(T_{i^*}^j(n-1)) + 1, & \text{for } X_{i^*} = 0 \end{cases} \tag{4.4.3}$$

2. **BLA-ρ^{RAND} Policy**

   The policy consists of the following three steps.

   - *Select channel to sense and access*: The procedure for selecting a channel and resolving collision is the same as UCB-ρ^{RAND}, only that we replace $\mathbf{I}^j(n)$ in UCB-ρ^{RAND} by $\mathbf{z}^j(n)$.

- *Update the hyper parameters*: Based on the selected channel availability, $\alpha^j_{i^*}(T^j_{i^*}(n))$ and $\beta^j_{i^*}(T^j_{i^*}(n))$ are updated as in (4.4.3).

$\boldsymbol{\theta}$-Dependent Adaptive Learning and Access Policies

1. **$\boldsymbol{\theta}$-Dependent Performance**

   Through our study, we find that the relative performance between the two distributed access policies, ρ^{RAND} and DLF, depends on the distribution of mean channel availabilities in $\boldsymbol{\theta}$. In particular, ρ^{RAND} policy outperforms DLF policy when θ_i's are close to each other; otherwise, DLF outperforms ρ^{RAND}. Such dependency can be explained through the collision resolving mechanism in two policies.

   For the DLF policy, each SU picks a channel with a rank in $\mathbf{I}^j(n)$ (or $\mathbf{z}^j(n)$) unique to this SU, *i.e.,* $r_i \neq r_j$, for $i \neq j$. If the ordering in $\mathbf{I}^j(n)$ truly reflects the ordering in $\boldsymbol{\theta}$, then each SU will access a unique channel among the top M ranked channels and no collision occurs. The DLF policy relies on self-correction to resolve the collision. That is, asymptotically, the mismatching between the ordering in $\mathbf{I}^j(n)$ and that in $\boldsymbol{\theta}$ is transient, due to exploration and exploitation trade-off in (3.4.2). The collision will resolve automatically once the ordering of the two quantities match again. Thus, DLF uses a *passive*-correcting mechanism for collision resolution.

   For the ρ^{RAND} policy, each SU picks a channel with a rank r_j in $\mathbf{I}^j(n)$ (or $\mathbf{z}^j(n)$) generated randomly, and resolves collision by redrawing r_j to reselect the channel. Different from DLF, the ρ^{RAND} policy actively resolves the collision through

redrawing r_j until no collision occurs. Thus, the ρ^{RAND} policy uses an *active*-correcting mechanism for collision resolution.

When θ_i's have similar values, their estimates are more prone to error, resulting in more frequent mismatch of the ordering $\mathbf{I}^j(n)$ and $\boldsymbol{\theta}$, and thus collisions. The passive-correcting mechanism in DLF works less efficient in this case than the active-correcting one in ρ^{RAND}, and the latter outperforms the former. On the other hand, when θ_i's are relatively spread, the passive-correcting mechanism is more efficient than the active-correcting mechanism, and DLF outperforms ρ^{RAND}.

The above analysis indicates that a given distributed access policy may only work well in certain type of $\boldsymbol{\theta}$ condition. In addition, as indicated in Section 4.4.2 that the underlying learning policies, UCB1 and BLA, also perform differently for different $\boldsymbol{\theta}$ distributions. It is thus desirable to design learning and access policies that work well regardless of primary channel mean availability $\boldsymbol{\theta}$ distribution. In the following, we propose an adaptive learning and access policy to aim at this goal.

2. **Dynamic Switching Learning and Access Policy**

We propose a dynamic switching learning and access policy (DSLA) that automatically switches between the underlying learning policies, UCB1 and BLA, as well as the access policies, ρ^{RAND} and DLF. The switching is based on a measurement on the closeness of the estimated mean channel availabilities $\{\hat{\theta}_i^j(T_i^j(n))\}$[1]. In the following, we propose a method to measure the closeness of $\{\hat{\theta}_i^j(T_i^j(n))\}$. Let $\left\{\hat{\theta}_{1^o}^j(T_{1^o}^j(n)), \cdots, \hat{\theta}_{N^o}^j(T_{N^o}^j(n))\right\}$ be the ordered statistics of $\{\hat{\theta}_i^j(T_i^j(n))\}$, with the subscript i^o denoting the quantity is the ith largest, *i.e.,* $\hat{\theta}_{1^o}^j(T_{1^o}^j(n)) > ... > \hat{\theta}_{N^o}^j(T_{N^o}^j(n))$. Denote

$$d_i^j(n) = \hat{\theta}_{i^o}^j(T_{i^o}^j(n)) - \hat{\theta}_{(i+1)^o}^j(T_{(i+1)^o}^j(n)), \tag{4.4.4}$$

for $i = 1, \cdots, N-1$. We define the closeness factor $\eta^j(n)$, for SU j at time slot n, as the percentage of $\hat{\theta}_{i^o}^j(T_{i^o}^j(n))$'s whose relative distances $d_i^j(n)$'s are within a predefined threshold ϵ, *i.e.,* let

$$i_{\max}^j(n) = \max\{i : d_i^j(n) \leq \epsilon, \;\; i = 1, \cdots, N-1\}, \tag{4.4.5}$$

then $\eta^j(n)$ is given by

$$\eta^j(n) \triangleq i_{\max}^j(n)/(N-1). \tag{4.4.6}$$

Smaller value of $\eta^j(n)$ indicates a smaller percentage of $\hat{\theta}_i^j(T_i^j(n))$'s having values close to each other. In this case, applying DLF policy for access is more efficient. Otherwise, ρ^{RAND} is more efficient. Thus, we use a threshold η_{th} to test against $\eta^j(n)$ to switch between DLF and ρ^{RAND}.

[1] Note that under the BLA learning algorithm, the mean channel availability θ_i can be estimated at SU j as $\hat{\theta}_i^j(T_i^j(n)) = \frac{\alpha_i^j(T_i^j(n))}{\alpha_i^j(T_i^j(n)) + \beta_i^j(T_i^j(n))}$.

Regarding which algorithm (UCB1 or BLA) to be used as the underlying distributed learning policy, from our extensive simulation studies, we observe that, for various distribution of $\{\theta_i\}$, when DLF policy is used, BLA outperforms UCB1 as the underlying distributed learning policy; and when ρ^{RAND} policy is used, UCB1 outperforms BLA. Therefore, we only need to dynamically switch between BLA-DLF and UCB-ρ^{RAND} based on $\eta^j(n)$. There is one exception to the above switching criterion: in our extensive experiments, we have also observed that when θ_i's are close to 1, even if they are similar to each other, BLA-DLF still outperforms UCB-ρ^{RAND}. Thus, we add an additional measure on the average value of θ_i's. Let

$$\overline{\hat{\theta}^j}(n) = \frac{1}{N}\sum_{i=1}^{N} \hat{\theta}_i^j(T_i^j(n)). \tag{4.4.7}$$

Let $\theta_{\text{th}} < 1$ be a predefined threshold close to 1. If $\overline{\hat{\theta}^j}(n) > \theta_{\text{th}}$, then regardless of the value of the closeness factor $\eta^j(n)$, BLA-DLF will be used. A summary of DSLA policy is given in Algorithm 5.

4.4.4 Simulation Results

We compare the performance under UCB-ρ^{RAND}, UCB-DLF, BLA-ρ^{RAND}, BLA-DLF policies, and the DSLA policy proposed in this chapter, in terms of normalized regret $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$, normalized with respect to $\log n$, over time n. We set $M = 4$, $N = 9$, and the thresholds $\eta_{\text{th}} = 0.2$, $\theta_{\text{th}} = 0.88$, $\epsilon = 0.02$.

In Figs. 4.10-4.12, we set $\boldsymbol{\theta} = [0.11, 0.12, \cdots, 0.19]^T$, $\boldsymbol{\theta} = [0.51, 0.52, \cdots, 0.59]^T$ and $\boldsymbol{\theta} = [0.91, 0.92, \cdots, 0.99]^T$, respectively, as the cases with similar mean availability across channels and low, moderate, and high availability, respectively. As we see,

Algorithm 5 DSLA Policy for SU j

1. **Input**:
   n: Current time slot; M: Number of secondary users; N: Number of channels;
   $\zeta_j(i,n)$: Collision indicator on channel i; r_j: Current rank;
   Closeness factor thresholds: ϵ, η_{th}

2. **Init**: $\alpha_i^j(0) \leftarrow 1$, $\beta_i^j(0) \leftarrow 1$ for $i = 1, \cdots, N$.
   $n \leftarrow 2$, $r_j \leftarrow 1$, $i^* \leftarrow N$, $\zeta_j(i,n) \leftarrow 0$ for $i = 1, ..., N$.

3. **Start Loop** $n \leftarrow n+1$.

   i) Obtain $\hat{\theta}_i^j(T_i^j(n))$, $i = 1, \cdots, N$;

   ii) Obtain $d_i^j(n)$ as in (4.4.4), $i = 1, \cdots, N-1$;

   iii) Obtain $\eta^j(n)$ as in (4.4.6);

   iv) **if** $\eta^j(n) < \eta_{\text{th}}$
   Run BLA-DLF
   else
   Compute $\overline{\hat{\theta}^j}(n)$ in (4.4.7);
   if $\overline{\hat{\theta}^j}(n) > \theta_{\text{th}}$
   Run BLA-DLF
   else
   Run UCB-ρ^{RAND}
   end if
   end if

   Stop Loop

in Fig. 4.10 and 4.11, our proposed DSLA policy achieves the same performance as UCB-ρ^{RAND} policy, which gives the best performance among all policies. Fig. 4.12 shows that our proposed DSLA policy outperforms UCB-ρ^{RAND} and UCB-DLF and approaches the BLA-DLF which gives the best performance among all policies.

In Fig. 4.13, we set $\boldsymbol{\theta} = [0.1, 0.2, \cdots, 0.9]^T$, *i.e.,* the mean channel availabilities are spread out. The BLA-DLF has the best performance among all policies in this $\boldsymbol{\theta}$ setting. Again, our proposed DSLA policy substantially outperforms UCB-ρ^{RAND}

and UCB-DLF, and approaches to that of BLA-DLF. In Fig. 4.14, we set $\boldsymbol{\theta}$ randomly as $\boldsymbol{\theta} = [0.21, 0.90, 0.76, 0.29, 0.83, 0.17, 0.68, 0.52, 0.39]$. The performance of our proposed DSLA policy again provides good performance among all policies, even though it is not the best.

4.4.5 Summary

In this chapter, we investigated into the problem on how the availability distribution of primary channels affect the performance of distributed learning and access policies. We first extended the recently proposed BLA algorithm to distributed online learning, and formed BLA-ρ^{RAND} and BLA-DLF learning and access policies modified from existing access policies. By analyzing the distributed access collision mechanism offered by the ρ^{RAND} and DLF policies, we identified how different mean channel availability distributions affect the effectiveness of each policy. Based on this, we developed dynamic switching mechanism for online learning and access algorithm (*i.e.,* DSLA) that adapts to different channel availability distribution conditions. The switching is based on our proposed closeness factor that determines which learning or access policies is most effective for a given primary channel condition. Simulation studies showed that our proposed DSLA policy is effective provides good performance for a wide range of θ_i's distributions.

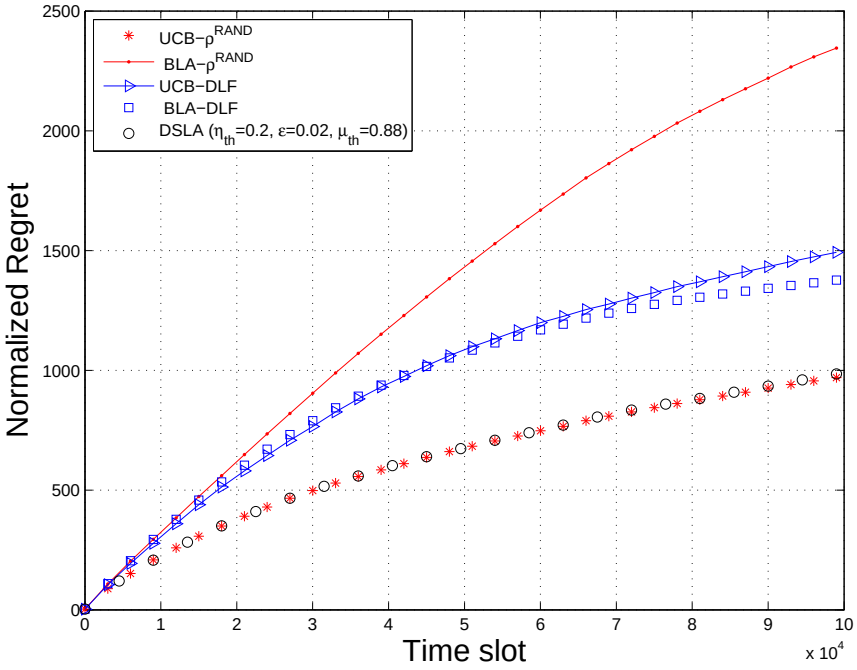

Figure 4.10: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ vs. time slot n. ($\boldsymbol{\theta} = [0.11, 0.12, ..., 0.19]$, M = 4, N = 9).

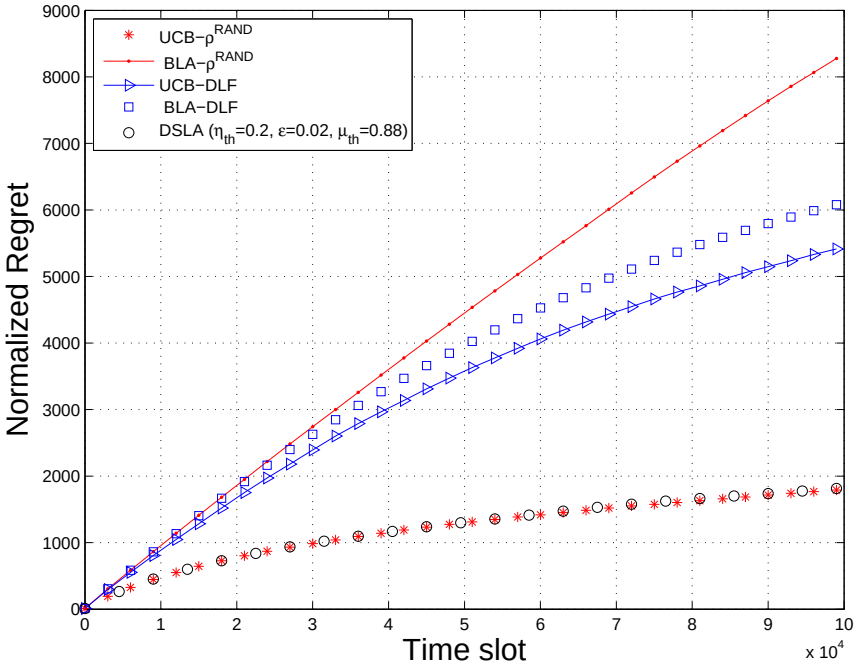

Figure 4.11: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ vs. time slot n. ($\boldsymbol{\theta} = [0.51, 0.52, ..., 0.59]$, M = 4, N = 9).

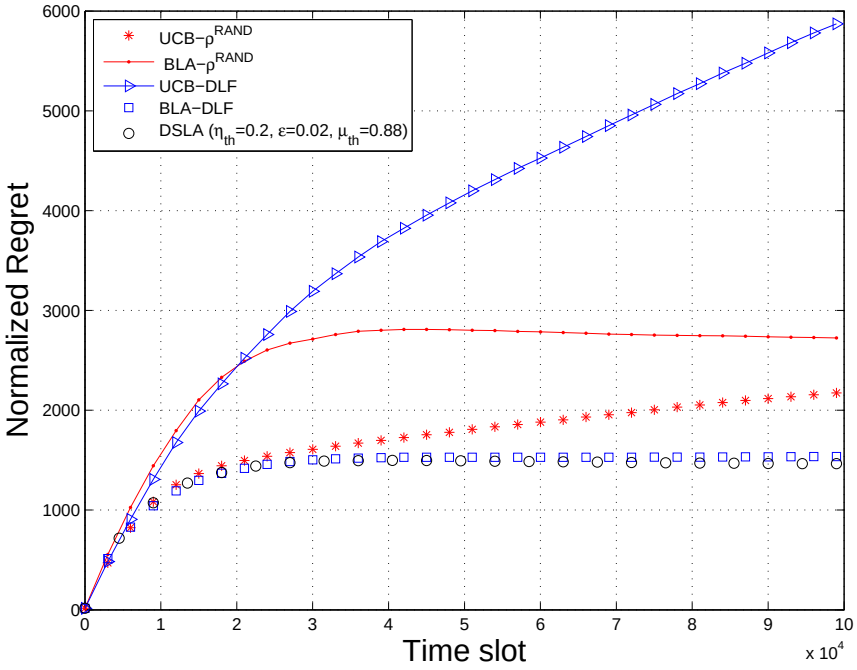

Figure 4.12: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ vs. time slot n. ($\boldsymbol{\theta} = [0.91, 0.92, ..., 0.99]$, M = 4, N = 9).

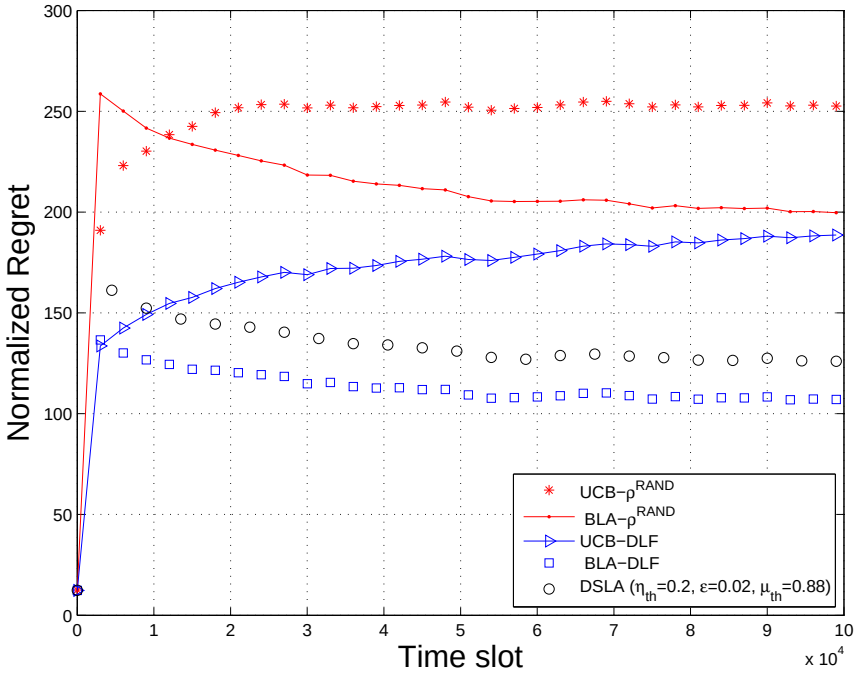

Figure 4.13: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ vs. time slot n. ($\boldsymbol{\theta} = [0.1, 0.2, ..., 0.9]$, M = 4, N = 9).

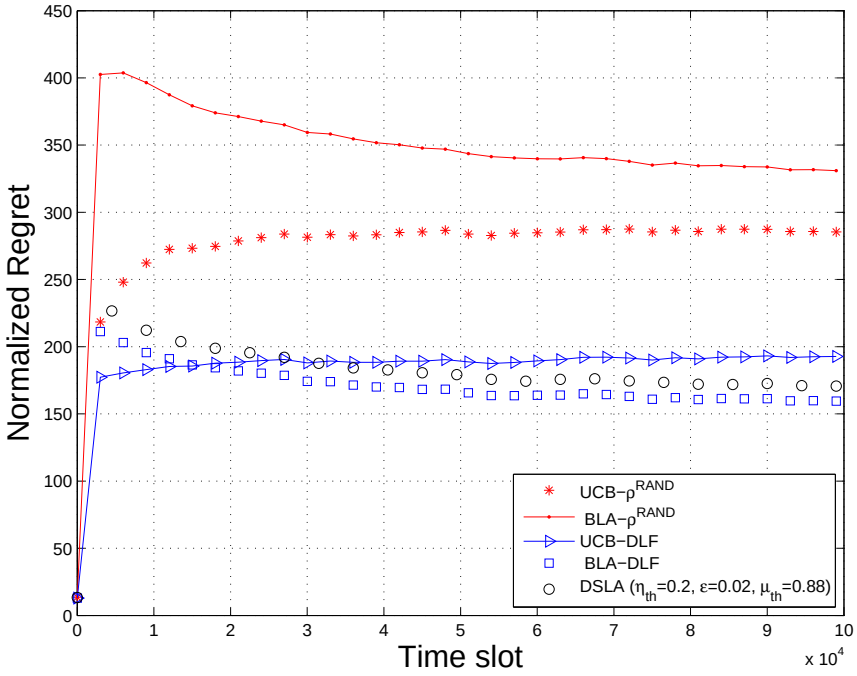

Figure 4.14: Normalized regrets $\frac{R(n,\boldsymbol{\theta},M)}{\log n}$ vs. time slot n. ($\boldsymbol{\theta} = [0.1, 0.2, 0.23, 0.94, 0.25, 0.8, 0.97, 0.98, 0.99]$, M = 4, N = 9).

Chapter 5

Dynamic Spectrum Access via Distributed Stochastic Learning Adaptive to Primary Channel Loading

5.1 Introduction

We consider a cognitive radio network with N independent primary channels for licensed users to access and M secondary users (SUs). Designing DSA mechanisms for efficient utilization of the spectrum is now considered as one of the main challenges cognitive radio networks face. The channel availability statistics of the primary network are typically unknown to the SUs. Through their limited spectrum sensing, the SUs search for the idle channels and make their own decisions based on their observation histories for channel access. Therefore, the challenges in designing a distributed policy for spectrum access among SUs involve not only online learning of the primary channel statistics using local sensing observations, but also designing an effective distributed mechanism for resolving collisions among SUs.

To overcome the above challenges, several decentralized learning and access policies have recently been developed [9, 10, 20–23, 93]. The works in [9, 10, 20] formulate the problem as extension version of the classical multi-armed bandit (MAB) problem [39–41] to decentralized one. In these policies, each SU relys on on its own sensing observation history to learn the most likely available primary channels to access. To resolve collision among SUs, these policies devise different mechanisms. It has been shown that the gap between the achieved throughput under a policy and that of the optimal one grows logarithmically over time in all these policies. Such a growth rate is considered to be order-optimal in terms of learning efficiency of a policy [39]. Furthermore, the relative performances of the access mechanisms devised for these policies are different for different distributions of primary channel availabilities across channels. As shown in [23], between the ρ^{RAND} policy [9] and the distributed learning with fairness (DLF) policy [10], the latter works more effectively than the former for dissimilar mean channel availabilities, but not so for channels with similar mean channel availabilities. Therefore, it is practically desirable to design learning and access policies that perform well for a wide range of primary mean channel availability distributions. Aiming at this goal, [23] propose a policy with an adaptive mechanism. This policy automatically switches between the ρ^{RAND} policy and the DLF policy, as well as two different underlying learning policies, by estimating the type of distribution of the primary channel availabilities. However, determining the switching threshold can be tricky, and the mechanism there can only switch between the two aforementioned policies.

5.1.1 Contributions

To address the above limitations, in our study, we formulate the distributed channel selection problem as a strategic game which is proved to be an exact potential game. We propose a distributed learning and access policy by applying stochastic learning automata (SLA) [64, 88]. Stochastic learning automata is used as a learning model in an unknown random environment. In DSA context, the cognitive radio network is considered as the unknown random environment in which the primary channel availability statistics are unknown to the SUs. In this context, the SUs are referred to the learning automata, adaptive decision making agents. The SUs try to learn the optimum channel selection through a series of interactions with the random environment. SLA aims at determining the optimal action out of a set of actions which are allowed.

In our proposed policy, we let each SU adjust its channel selection through learning from its collision history. Specifically, each SU learns from its own sensing history about the most likely available channels among N channels, and probabilistically selects one of them to access. The channel selection probability is then updated based on the collision events.

Our proposed adaptive policy utilizes the two following types of underlying distributed learning: 1. Learning from SU's own sensing history on the primary channel availability, and 2. Learning from SU's own collision history to adjust its channel selection among SUs for collision avoidance. Through these two learning mechanisms in parallel, each SU's channel selection can adapt to different type of channel availability distributions across channels. We show that both the ρ^{RAND} and the DLF policies can be viewed as special cases of our proposed adaptive policy, by pre-setting the channel

selection probabilities. Numerical results show that our proposed policy outperforms these existing policies in various types of mean channel availabilities across primary channels.

5.2 Network Model

We consider a cognitive radio network consists of M secondary users (SUs) independently searching for idle channels among the N channels which are licensed to a slotted primary network, where $M \leq N$. We assume that M is known to the SUs. Let $X_i(n)$ denote the availability state of the ith channel in the primary network at time slot n, where $X_i(n) = 1$ if the ith channel is available and 0 otherwise. We assume $X_i(n)$ evolves as an i.i.d. Bernoulli random process over n, with the mean $\theta_i = E[X_i(n)] \in [0, 1]$, *i.e.,* $X_i(n) \sim$ Bernoulli(θ_i), $\forall n$ and $i = 1, \cdots, N$. We assume θ_i's are distinct from each other and are unknown to the SUs. Let $\boldsymbol{\theta} \triangleq [\theta_1, \theta_2, ..., \theta_N]$ denote the mean channel availability vector.

At the beginning of time slot n, each SU selects a channel to sense, and if available, the SU then will access the channel. In this work, we assume perfect channel sensing at all SUs. Each SU tries to learn the unknown mean availabilities of the primary channels $\boldsymbol{\theta}$ over time using its own sensing outcomes and observation history. For channel access among SUs, if same channel is selected by more than one SU to be accessed, then these SUs' transmissions fail, resulting in zero throughput for each of them. Otherwise, if the channel is selected by only one SU, then the sole SU who accesses the channel will receive a unit throughput.

Regret is a common metric used to measure the throughput loss of a given policy under learning for any learning and access policy [39]. It is defined as the difference

over throughput between the ideal scenario and a given policy, *i.e.*,

$$R(n) \triangleq n \sum_{k=1}^{M} \theta_{k^*} - \sum_{i=1}^{N} \sum_{j=1}^{M} \theta_i E[S_i^j(n)] \tag{5.2.1}$$

where $S_i^j(n)$ denotes the number of times, up to current slot n, that SU j is the sole user to sense channel i, and k^* represents index such that θ_{k^*} is the kth-highest valued element in $\boldsymbol{\theta}$.

Our design objective is to devise a distributed access policy that minimizes the regret, where each SU makes its access decision based on its own estimate of the primary channel conditions.

5.3 A Distributed Adaptive Learning and Access Policy

5.3.1 Distributed Learning and Sensing of Primary Channels

A sensing policy is designed to learn the unknown mean channel availabilities $\boldsymbol{\theta}$. In [44], the UCB1, an online learning algorithm of the unknown parameters (*e.g.*, channel availabilities in dynamic spectrum access), is proposed for the single user case. It is an index-based learning policy. An index is computed for each channel and is used to rank the channel. At each time slot n, the user then selects the highest-ranked channel. In [39, 44], it is shown that UCB1 is an order-optimal learning algorithm in the sense that the algorithm achieves the logarithmic growth of the regret.

For dynamics spectrum access, decentralized access policies proposed in [9, 10, 20] extend the UCB1 algorithm to the distributed case for the underlying learning of

channel availability statistics. Let $T_i^j(n)$ denote the number of times that the SU j senses channel i up to time slot n. If SU j selects channel i to sense at time slot n, then it obtains the value of $X_i(n)$ and records it as $X_i^j(T_i^j(n))$. Let $\mathbf{X}_i^j(n) \triangleq [X_i^j(1), \cdots, X_i^j(T_i^j(n))]$ be the vector holding the sensing observation history of channel i up to time slot n at SU j. Using $\mathbf{X}_i^j(n)$, SU j estimates θ_i of channel i at time n as in (3.4.1). Define a ranking-index for channel i at SU j as in (3.4.2). It will be used for ranking the channels by SU j. Specifically, each SU j computes a ranking-index vector $\mathbf{I}^j(n) \triangleq [I_1^j(n), \cdots, I_N^j(n)]$ based on its own observation history. Then, it will select the channel whose index value is the kth-highest in $\mathbf{I}^j(n)$ to access, for some $0 \leq k \leq M$.

Distributed learning policies [9,10,20] propose different mechanisms for access coordination among SUs to choose different channels among the first M-highest ranked channels.

5.3.2 Distributed Access: Stochastic Learning in Secondary Access Environment

As shown in [23], the existing distributed access policies may work well in one type of mean channel availability distribution in $\boldsymbol{\theta}$ but not in other cases. In other words, the effectiveness of a proposed policy will be impacted by different channel availability distributions, resulting in different relative performance. Designing learning and access policies that work well for a wide range of mean channel availability distribution $\boldsymbol{\theta}$ of the primary channels is practically desirable. To design such a policy, we propose an adaptive learning and access policy based on the idea of SLA [64] to adapt each SU's channel selection through the learning from its collision history. SLA is used

as a learning model in an unknown random environment. For spectrum access, the primary network can be considered as the unknown random environment. Each SU is considered as a learning automaton for adaptive decision making.

Define the *M-best* channels as those channels with θ_i values being among the M highest ones in $\boldsymbol{\theta}$, and the *estimated M-best* channels as those channels whose ranking-indexes $I_i^j(n)$'s are among the M highest in $\mathbf{I}^j(n)$. Denote $\sigma_{j,r}$ the index of the channel who is ranked rth in $\mathbf{I}^j(n)$ for SU j. Let $\mathcal{C}_M^j(n)$ be the set of indexes of the estimated M-best channels for SU j at time slot n, given by

$$\mathcal{C}_M^j(n) = \{\sigma_{j,1}(n), \cdots, \sigma_{j,M}(n)\}. \tag{5.3.1}$$

We let each SU j probabilistically choose a channel. Let $p_i^j(n)$ denote the probability of selecting channel i by SU j at time slot n. Denote the channel selection probability distribution vector for SU j by

$$\mathbf{p}^j(n) \triangleq [p_1^j(n), \cdots, p_N^j(n)]^T \tag{5.3.2}$$

where entries of $\mathbf{p}^j(n)$ satisfies $\sum_{i=1}^N p_i^j(n) = 1$. At the start of the process, SU j's channel selection probability distribution is uniformly initialized $\mathbf{p}^j(0) = [1/N, \cdots, 1/N]$.

Channel Selection

At time slot n, SU j selects a channel in the set of its estimated M-best channels $\mathcal{C}_M^j(n)$. To do this, we re-normalize the channel selection probabilities of these channels, *i.e.,* p_i^j, for $i \in \mathcal{C}_M^j(n)$. Define $\bar{p}_i^j(n)$ as the re-normalized probability of the channel i, for $i \in \mathcal{C}_M^j(n)$. It is computed as

$$\bar{p}_i^j(n) = \frac{p_i^j(n)}{\sum_{i \in \mathcal{C}_M^j(n)} p_i^j(n)}, \quad \text{for } i \in \mathcal{C}_M^j(n). \tag{5.3.3}$$

Let $\bar{\mathbf{p}}^j(n) \triangleq [\bar{p}^j_{\sigma_{j,1}}(n), \cdots, \bar{p}^j_{\sigma_{j,M}}(n)]$ be the re-normalized channel selection probability vector for SU j for its estimated M-best channels. SU j then selects the rank $r_j \in \{1, \cdots, M\}$ with the probability distribution $\bar{\mathbf{p}}^j(n)$, and chooses the corresponding channel σ_{r_j} (*i.e.,* the r_jth-highest ranked in $\mathcal{C}^j_M(n)$ based on $\mathbf{I}^j(n)$). SU j then senses the channel and accesses it if it is available.

In the next time slot $(n+1)$, if no collision occurs, SU j will maintain the rank selection r_j, and select a channel σ_{j,r_j} again in the updated set $\mathcal{C}^j_M(n+1)$ of the estimated M-best channels. Otherwise, SU j will redraw the rank $r_j \in \{1, \cdots, M\}$ with $\bar{\mathbf{p}}^j(n+1)$, and select channel σ_{j,r_j}.

Channel Selection Probability Update

Each SU j uses an acknowledgement for collision feedback. Let $\zeta^j_i(n) \in \{0, 1\}$ denote the acknowledgment of SU j's collision on channel i, where $\zeta^j_i(n) = 1$ denotes the collision event and 0 otherwise. Based on the collision model, for SU j, we define $\Upsilon^j_i(n)$ as the reward in terms of throughput by accessing channel i at time slot n, given by

$$\Upsilon^j_i(n) = \begin{cases} 1, & \text{if } X_i(n) = 1, i = \sigma_{j,r_j}(n), \text{and } \zeta^j_i(n) = 0; \\ 0, & \text{otherwise} \end{cases} \tag{5.3.4}$$

Define $\Upsilon^j(n)$ as the reward received by SU j by accessing the primary network at time slot n, given by

$$\Upsilon^j(n) = \sum_{i \in \mathcal{C}^j_M(n)} \Upsilon^j_i(n) = \Upsilon^j_{\sigma_{j,r_j}}(n). \tag{5.3.5}$$

Since each SU at most access one channel at a time, $\Upsilon^j(n) \in \{0, 1\}$. Also, due to possible collision, $\Upsilon^j(n)$ is random.

Based on the reward $\Upsilon^j(n)$, SU j updates its channel selection probability distribution $\mathbf{p}^j(n)$ to $\mathbf{p}^j(n+1)$ according to the following SLA based rule:

$$p_i^j(n+1) = p_i^j(n) + b\Upsilon^j(n)(1 - p_i^j(n)), \quad \text{for } i = \sigma_{j,r_j}(n) \tag{5.3.6}$$

$$p_i^j(n+1) = p_i^j(n) - b\Upsilon^j(n)p_i^j(n), \quad \text{for } i \neq \sigma_{j,r_j}(n) \tag{5.3.7}$$

where $b \in (0,1)$ is the updating step size. From (5.3.5)-(5.3.7), we verify that $\sum_i^N p_i^j(n+1) = 1$. Note that the above SLA based updating rule adaptively adjusts the channel selection probabilities based on the reward from the access attempt. When $\Upsilon^j(n) = 0$ (either due to the primary channel not being available or collision), $p_i^j(n)$ to $p_i^j(n+1)$ remains unchanged, $\forall i$. When SU j accessing channel σ_{j,r_j} is successful, $p_i^j(n+1)$ will be increased, for the accessed channel $i = \sigma_{j,r_j}$, while that for the rest channels will be decreased. A summary of the proposed algorithm is given in Algorithm 6 (see Fig. 5.1).

5.3.3 $\boldsymbol{\theta}$-Dependent Channel Selection Adaptation

The proposed sensing and access policy consists of the two following underlying distributed learning algorithms: 1. Distributed learning of the primary channel availability $\boldsymbol{\theta}$ based on each SU's own sensing history. This learning mechanism ensures that each SU to select among the most available channels for access to improve throughput. 2. Distributed learning of channel selections among SUs, through collision events, to automatically adjust and orthogonalize the channel selections among SUs. This is reflected in the value of $\mathbf{p}^j(n)$ for each SU j as it converges. As shown in simulations, each SU j will select a particular channel with probability approaches to 1, and the selections among SUs are orthogonalized.

Having a closer look at the learning of channel selection, we observe that the channel selection probability updates are impacted by the accuracy of the estimate $\hat{\boldsymbol{\theta}}$. For $\{\theta_i\}$, especially for those of M-best channels, being spread out, the learning is relatively more accurate over time. This means a given rank r_j will match the rank of the actual channel availability more accurately. Thus, if no collision, SU j's rank selection r_j and σ_{r_j} will mostly remain the same and unchanged, resulting in quick convergence of channel selection probability $\mathbf{p}^j(n)$ over time to a pure probability vector of a specific rank r_j^o (e.g. with probability approaching 1 to select a rank r_j^o). The access policy effectively converges to a policy where each SU selects a channel with a fixed (orthogonalized) rank to access.

On the other hand, if channels are similar in mean availability, the learning of $\boldsymbol{\theta}$ is relatively inaccurate and slow over time. Each SU may rank channels differently, due to the mismatch of rank r_j and the true rank of the channel availability. This results in a more collision-prone scenario among SUs, and in return, a re-selection of the rank r_j for SU j. Thus, the channel selection probability will be changed slowly from the initial uniform distribution before the estimate $\hat{\boldsymbol{\theta}}$ becomes more accurate. The benefit of this slow convergence is that during this process, each SU actively re-selects a channel to access, resulting in proactively resolving collision among users to reduce the throughput loss. Using a pure probability vector for each SU during this time may result in more collisions without actively re-selecting a different channel to access. Thus, through such an adaptive adjustment of selection probability distribution, the proposed algorithm nicely adjusts its "collision resolution strategy" based on the mean channel availability distribution across primary channels.

5.3.4 Relation to Existing Distributed Access Policies

The ρ^{RAND} policy [9] and the DLF policy [10], are two MAB based existing distributed access policies which use different mechanisms for access coordination. The access mechanisms ensure that SUs select different channels among the estimated M-best channels. Both of them have been shown to be order-optimal in the growth rate of regret.

Based on our analysis in Section 5.3.3, we observe that both ρ^{RAND} and DLF can be considered as a special case of the proposed adaptive policy in Algorithm 6, with the channel selection probability distribution $\bar{\mathbf{p}}^j(n)$ being either uniform or pure (*i.e.,* '0' or '1' value).

[23] analyzes the two policies and observes that these two policies perform differently for different $\boldsymbol{\theta}$ distribution. The ρ^{RAND} policy performs better when the values of θ_i's are similar, and the DLF policy performs better when the values of θ_i's are relatively spread out. In response to $\boldsymbol{\theta}$ distribution, our proposed policy provides an adaptive approach. As our numerical results show, the proposed policy outperforms both policies in a wide range of $\boldsymbol{\theta}$ distributions.

5.4 Game Theoretic Formulation

In the following, we formulate the distributed channel selection and access problem previously considered as a *non-cooperative strategic game*[1] [96], where the SUs are considered as the players, and the licensed primary channels are the possible actions that the SUs may take.

[1] A strategic game is a game in which the interaction of the players (decision-makers) are considered where each decision-maker selects its action once , and these actions are taken simultaneously.

Specifically, let $\mathcal{S} = \{1, ..., M\}$ denote the set of SUs and $\mathcal{C} = \{1, ..., N\}$ denote the set of channels for the SUs to select (actions); the action can be a primary channel $i \in \{1, \cdots, N\}$. Let $\mathcal{A}_j(n) \subseteq \mathcal{C}$ denote the set of actions that SU j can take from at time slot n and $a_j(n)$ denote the action taken by SU j at time slot n. Define $a_{-j}(n)$ as the set of actions taken by SU j's opponents as $a_{-j}(n) \triangleq \{a'_j(n) : j' \in \mathcal{S}\backslash\{j\}\}$. Denote $u_j(a_j, a_{-j}; n)$ the payoff of SU j upon taking action $a_j(n) \in \mathcal{A}_j(n)$ while others taking $a_{-j}(n)$. Then, $\mathcal{G}_p \triangleq \{\mathcal{S}, \{\mathcal{A}_j(n)\}_{j\in\mathcal{S}}, \{u_j(a_j, a_{-j}; n)\}_{j\in\mathcal{S}}\}$ forms a game [97]. Note that after each SU j determines its action $a_j = q$, it will make access decision depending on the channel availability, i.e. $X_q = 1$.

Let $m_a(n)$ denote the number of SUs taking the same action $a(n)$ at time slot n. From the acknowledgement $\zeta_a^j(n)$ for SU j defined in Section 5.3.2, we have the following equivalence

$$m_a(n) = 1 \Leftrightarrow \zeta_a^j(n) = 0,$$
$$m_a(n) > 1 \Leftrightarrow \zeta_a^j(n) = 1.$$

We set the action set for SU j as $\mathcal{A}_j(n) = \mathcal{C}_M^j(n)$, *i.e.,* SU j can select one of its estimated M-best channels. Then, accessing channel i (*i.e.,* $i = \sigma_{j,r_j}(n)$) can be viewed as taking action $a_j(n)$ where $X_i(n) = 1$. Therefore, the reward $\Upsilon_i^j(n)$ in (5.3.4) can be interpreted as the reward for SU j taking action a, re-expressed as

$$\Upsilon_a^j(n) = \begin{cases} 1, & \text{if } a(n) = a_j(n), m_a(n) = 1, \text{ and } X_{a_j}(n) = 1; \\ 0, & \text{otherwise.} \end{cases} \tag{5.4.1}$$

Similarly, the reward $\Upsilon^j(n)$ in (5.3.5) can be re-expressed by

$$\Upsilon^j(n) = \sum_{a\in\mathcal{A}_j(n)} \Upsilon_a^j(n) = \Upsilon_{a_j}^j(n). \tag{5.4.2}$$

We define the payoff of SU j as its expected throughput achieved from accessing the primary network, defined as

$$\begin{aligned} u_j(a_j, a_{-j}; n) &\triangleq E[\Upsilon^j(n)|a_j(n), a_{-j}(n)] \\ &= \psi_{a_j}(m_{a_j}(n)) \end{aligned} \tag{5.4.3}$$

where the expectation is taken over the randomness of the channel availability state, and

$$\psi_i(k) \triangleq \begin{cases} \theta_i, & \text{if } i \in \{1, ..., N\}, \text{and } k = 1; \\ 0, & \text{otherwise.} \end{cases} \tag{5.4.4}$$

For (5.4.3), this means

$$\psi_a(m_a(n)) = \begin{cases} \theta_a, & \text{if } a(n) \neq 0, m_a(n) = 1; \\ 0, & \text{otherwise.} \end{cases} \tag{5.4.5}$$

5.4.1 Exact Potential Game

A game is called a *potential game* [98, 99], where the incentives of all players of the game for changing their actions can be reflected by a function which is called a *potential function*. Showing the existence of a potential function in a game is sufficient to prove the game being a potential game.

Let $\mathcal{P}(a_j, a_{-j}; n)$ denote the potential function of a game if SU j takes action $a_j(n) \in \mathcal{A}_j(n)$ and $\mathcal{P}(\tilde{a}_j, a_{-j}; n)$ denote the potential function if SU j takes action $\tilde{a}_j(n) \in \mathcal{A}_j(n)$. An *exact potential game* [98,99] is defined as a game where there exists a potential function if player j, changes its action from $a_j(n)$ to $\tilde{a}_j(n)$, the deviation

in the payoff of the player j is reflected by deviation in the potential function, *i.e.*

$$\begin{aligned}&\mathcal{P}(\tilde{a}_j, a_{-j}; n) - \mathcal{P}(a_j, a_{-j}; n)\\ &= u_j(\tilde{a}_j, a_{-j}; n) - u_j(a_j, a_{-j}; n).\end{aligned} \tag{5.4.6}$$

Property of NE

One of the most important properties of the exact potential game is that it can achieve at least one pure-strategy NE [98].

Pure strategy: In a stochastic game, a player selects an action with certain probability. The channel selection probability distribution vector $\bar{\mathbf{p}}^j(n)$ below (5.3.3) defines a strategy for SU j. Any unit probability distribution vector (*i.e.,* only one entry with 1 and the rest 0) represents a pure strategy. Other non-unit vectors represent mixed strategies.

Pure-strategy NE: At time slot n, an action profile $\mathbf{a}^o(n) = [a_1^o(n), ..., a_M^o(n)]$ is called a pure strategy NE for $\mathcal{G}_p$[2], if and only if no SU can obtain a higher payoff by deviating unilaterally from this profile. That is, let $m_{a_j^o}(n)$ and $m_{a_j}(n)$ be the number of SUs selecting actions $a_j^o(n)$ and $a_j(n)$, respectively, at time slot n. If SU j changes its action from $a_j^o(n)$ to $a_j(n)$ while others keep their actions unchanged, the number of SUs selecting actions $a_j(n)$ becomes $m_{a_j}(n) + 1$. Then, the following property holds

$$\begin{aligned}&\psi_{a_j^o}(m_{a_j^o}(n)) \geq \psi_{a_j}(m_{a_j}(n) + 1),\\ &\qquad\qquad \forall a_j(n) \in \mathcal{A}_j(n) \backslash \{a_j^o(n)\},\ \forall j \in \mathcal{S}.\end{aligned} \tag{5.4.7}$$

[2]In other words, player j taking an action $a_j^o(n) \in \mathbf{a}^o(n)$ with probability 1.

5.4.2 $\mathcal{G}_p$ as an Exact Potential Game

We now show that with the payoff defined in (5.4.3), there exists a potential function with property as in (5.4.6), and the game $\mathcal{G}_p$ is an exact potential game.

Proposition 3. *The channel selection and access game $\mathcal{G}_p$ is an exact potential game.*

Proof. See Appendix B.1.

By proposition (3) and the property of the exact potential game, it follows that the game $\mathcal{G}_p$ can at least achieve one pure strategy NE point. In the next section, we analyze the convergence behavior of our proposed algorithm in Section 5.3.2 to the NE points of the game $\mathcal{G}_p$.

5.5 Convergency of the proposed algorithm towards pure Strategy NE of the game $\mathcal{G}_p$

In this section, we show that the distributed adaptive learning and access policy (Algorithm 6) proposed in Section 5.3 converges to the pure strategy NE of the game $\mathcal{G}_p$.

A multi-person stochastic game is considered in [88], where the convergence of a proposed SLA-based algorithm to NE is analyzed. Although our game $\mathcal{G}_p$ is different from the game defined there, we can adopt the approach in [88] to investigate the convergence of our algorithm. That is, we use the solution of ordinary differential equation (ODE) to analyze and understand the long term behaviour of the channel selection probability matrix $\mathbf{P}(n)$.

Define $\mathbf{P}(n) \triangleq [\mathbf{p}^1(n), \mathbf{p}^2(n), ..., \mathbf{p}^M(n)]$ as the $N \times M$ channel selection probability matrix, where $\mathbf{p}^j(n)$ is given in (5.3.2). Define $\mathbf{a}(n) \triangleq [a_1(n), a_2(n), ..., a_M(n)]$ as the channel selection (action) profile and $\boldsymbol{\Upsilon}(n) \triangleq [\Upsilon^1(n), \Upsilon^2(n), ..., \Upsilon^M(n)]$ the reward profile at time slot n. Then, (5.3.6) and (5.3.7) can be combined to be re-expressed as

$$\mathbf{P}(n+1) = \mathbf{P}(n) + bG(\mathbf{P}(n), \mathbf{a}(n), \boldsymbol{\Upsilon}(n)) \tag{5.5.1}$$

where $G(\mathbf{P}(n), \mathbf{a}(n), \boldsymbol{\Upsilon}(n))$ is specified by the updating rule in (5.3.6) and (5.3.7), and is a function of $\mathbf{P}(n)$, $\mathbf{a}(n)$, and $\boldsymbol{\Upsilon}(n)$. Define $f(\mathbf{P})$ as the conditional expectation of $G(\cdot)$ given $\mathbf{P}(n) = \mathbf{P}$ as

$$f(\mathbf{P}) = E[G(\mathbf{P}(n), \mathbf{a}(n), \boldsymbol{\Upsilon}(n))|\mathbf{P}(n) = \mathbf{P}].$$

First, we show the limiting behavior of the sequence of $\{\mathbf{P}(n)\}$ in the following proposition.

Proposition 4. *As the step size $b \to 0$, the sequence of the piecewise-constant interpolation of $\{\mathbf{P}(n)\}$ defined by $\tilde{\mathbf{P}}(t) = \mathbf{P}(n)$, for $t \in [nb, (n+1)b]$, converges weakly to the solution $\mathbf{X}^*(t)$ of the ODE defined by*

$$\frac{d\mathbf{X}(t)}{dt} = f(\mathbf{X}(t)),\ \mathbf{X}(0) = \tilde{\mathbf{P}}(0) = \mathbf{P}(0) \tag{5.5.2}$$

where $\mathbf{P}(0)$ is the initial channel selection probability matrix.

Proof. The result is a direct application of [88, Theorem 3.1]. □

Proposition 4 indicates that for sufficiently small step size, the long-term behavior of the sequence $\{\mathbf{P}(n)\}$ follows the trajectory $\mathbf{X}^*(t)$ of the ODE in (5.5.2). This allows us to use the stability properties of ODE to analyze the algorithm. The following

proposition captures the relationship between the stable stationary points of the ODE and the Nash equilibria of $\mathcal{G}_p$.

Proposition 5. *All the stable stationary points of the ODE in (5.5.2) are the Nash equilibria of the game $\mathcal{G}_p$. All the pure-strategy Nash equilibria of $\mathcal{G}_p$ are the asymptotically-stable stationary points of the ODE in* (5.5.2).

Proof. The result follows immediately from [88, Theorem 3.2]. □

Based on Propositions 4 and 5, we now characterize the long term behavior of our proposed algorithm.

Proposition 6. *Assuming a sufficiently small step size b, our proposed distributed adaptive learning and access algorithm converges to a pure strategy Nash equilibrium of the game $\mathcal{G}_p$.*

Proof. See Appendix B.2.

5.6 Simulation Results

In this section, we study the performance of the proposed policy in Algorithm 6. We assume a cognitive radio network with $M = 4$ SUs independently searching among $N = 9$ primary channels. At each time slot n, the channel availability $X_i(n)$ follows i.i.d. Bernoulli random process, for $i = 1, \cdots, N$ with mean θ_i, unknown to the SUs. All simulations are performed for 200 Monte Carlo runs.

All the case examples of the mean channel availability vector $\boldsymbol{\theta}$ considered in our simulations in this work are listed in Table 5.1. Cases 1 to 3 represent three different types of primary channel traffic loads for $N = 9$ and $M = 4$. Case 1 represents a

scenario of dissimilar channels, where the channel traffic loads are evenly spread out. Case 2 represents a scenario where the average loads across different channels are random. On the contrary, case 3 shows a case where the average loads on all channels are considered to be similar.

To measure the performance of our proposed algorithm, we use normalized regret, defined as $R(n)/\log n$. The regret measures the convergence rate of the achieved throughput. In implementing Algorithm 6, a threshold p_{th} is used to check the convergence of the channel selection probability vector $\mathbf{p}^j(n)$ to a pure probability vector. If $p^j_{i^o} > p_{\text{th}}$, for some i^o, then we set $p^j_{i^o} = 1$ and $p^j_i = 0$ for $i \neq i^o$. In the simulations, we set $p_{\text{th}} = 0.9$.

5.6.1 Convergence behavior of the Channel Selection Probabilities

We first show in Figs. 5.2 to 5.5 the convergence behavior of channel selection probability $p^j_i(n)$ on each channel i, $i \in \{1, \cdots, N\}$, for SU j, $j \in \{1, \cdots, M\}$, under our proposed policy. We set the mean channel availability vector $\boldsymbol{\theta} = [0.1, 0.2, ..., 0.9]$ as in case 1. As it can be seen from Fig. 5.2, the selection probability for SU 1 on channel 6 converges to 1, while the selection probabilities on the rest of channels converge to 0, *i.e.* the selection probability for SU 1 evolves from $\mathbf{p}^1(n) = [\frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}]$ to $\mathbf{p}^1(n) = [0, 0, 0, 0, 0, 1, 0.0.0]$. Thus, the SU 1 asymptotically selects channel 6 ($\theta_6 = 0.6$, among the M-best channels) with probability approaching to 1.

Fig. 5.3 shows that the selection probability for SU 2 on channel 7 converges to 1, while the selection probabilities on the rest of channels converge to 0, *i.e.* the selection probability for SU 2 evolves from $\mathbf{p}^2(n) = [\frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}]$ to

N	M	Case	$\boldsymbol{\theta}$
9	4	1	[0.1, 0.2, ... , 0.9]
		2	[0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7]
		3	[0,51, 0.52, ... , 0.59]

Table 5.1: Simulation cases of mean channel availability $\boldsymbol{\theta}$.

N	M	SU	Selected Channel
9	4	1	6
		2	7
		3	8
		4	9

Table 5.2: Channel asymptotically selected by probability one, $\boldsymbol{\theta}$: case 1.

$\mathbf{p}^2(n) = [0, 0, 0, 0, 0, 0, 1, 0, 0]$. Therefore, the SU 2 asymptotically selects channel 7 ($\theta_7 = 0.7$, among the M-best channels) with probability approaching to 1.

The evolution of the channel selection probabilities for SU 3 is shown in Fig. 5.4. The selection probability for SU 3 evolves from $\mathbf{p}^3(n) = [\frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}]$ to $\mathbf{p}^3(n) = [0, 0, 0, 0, 0, 0, 0, 1, 0]$, *i.e.* the SU 3 asymptotically selects channel 8 ($\theta_8 = 0.8$, among the M-best channels) with probability approaching to 1.

The similar behavior is observed for SU 4, where it asymptotically selects channel 9 which is also among the M-best channels and is orthogonal to other SUs to access. The selection probability for SU 4 evolves from $\mathbf{p}^4(n) = [\frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}]$ to $\mathbf{p}^4(n) = [0, 0, 0, 0, 0, 0, 0, 0, 1]$. These observations are depicted in Table 5.2.

We also show in Figs. 5.6 to 5.9 the time trajectories of channel selection probability $p_i^j(n)$ on each channel i for the SUs under our proposed policy. We set $\boldsymbol{\theta} = [0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7]$ as in Case 2. As it can be seen, for *i.e.,* SU 1 the selection probability on channel 5 converges to 1, while the selection

N	M	SU	Selected Channel
9	4	1	5
		2	9
		3	6
		4	8

Table 5.3: Channel asymptotically selected by probability one, $\boldsymbol{\theta}$: case 2.

probabilities on the rest of channels converge to 0. Thus, the SU 1 asymptotically selects channel 5 ($\theta_5 = 0.67$, among the M-best channels) with probability approaching to 1. The similar behavior is observed for all other SUs, where each SU asymptotically selects one of the M-best channels orthogonal to other SUs to access as depicted in Table 5.3.

We finally show in Figs. 5.10 to 5.13 the time trajectories of channel selection probability $p_i^j(n)$ on each channel i for the SUs under our proposed policy. We set $\boldsymbol{\theta} = [0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59]$ as in case 3. As it can be seen, for *i.e.,* SU 1 the selection probability on channel 6 converges to 1, while the selection probabilities on the rest of channels converge to 0. Thus, the SU 1 asymptotically selects channel 6 ($\theta_5 = 0.56$, among the M-best channels) with probability approaching to 1. The similar behavior is observed for all other SUs, where each SU asymptotically selects one of the M-best channels orthogonal to other SUs to access as depicted in Table 5.4.

As it can be seen from the simulation results explained above, our observations are intuitively correct. The distributed learning of the primary channel availability $\boldsymbol{\theta}$ based on each SU's own sensing history ensures that each SU selects among the most available channels for access. The distributed learning of channel selections

N	M	SU	Selected Channel
9	4	1	6
		2	8
		3	9
		4	7

Table 5.4: Channel asymptotically selected by probability one, $\boldsymbol{\theta}$: case 3.

among SUs, through collision events, automatically adjusts and orthogonalizes the channel selections among SU which is reflected in the value of $\mathbf{p}^j(n)$ for each SU j as it converges.

5.6.2 Comparison with Existing Access Policies

We further compare the performance of our proposed policy with the following three existing access policies, *i.e.,* ρ^{RAND}, DLF, and another SLA-based policies proposed in [93][3]. In Fig. 5.14, we compare the normalized regret under our proposed policy with that of three aforementioned policies. The same $\boldsymbol{\theta}$ as in Fig. 5.2 is used, which represents the case where channels are dissimilar in terms of mean channel availabilities. In Fig. 5.15, the same comparison is performed where channel availabilities are set randomly with $\boldsymbol{\theta} = [0.3, 0.34, 0.5, 0.6, 0.67, 0.91, 0.2, 0.8, 0.7]$ as in case 2. We also examine the case where channels are similar in their mean availabilities. We set $\boldsymbol{\theta} = [0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59]$ as in case 3, and show the comparisons of the normalized regret over time slot n among the four access policies in Fig. 5.16. From Figs. 5.14 to 5.16, we see that, as discussed in Section 5.3.4, the relative performance of ρ^{RAND} versus DLF depends on the distribution in $\boldsymbol{\theta}$. Nonetheless,

[3]The policy proposed in [93] does not take the channel availability into account, thus treats every channel in the same manner.

our proposed policy significantly outperforms these two policies in all three types of $\boldsymbol{\theta}$ values.

As it can be seen, our proposed policy also significantly outperforms the policy in [93] in all cases except for the case with similar mean channel availabilities θ_i's. In [93], the secondary users do not distinguish different primary channels and can select any one of those channels. When θ_i's are dissimilar, using our policy, SUs will avoid those channels with low mean channel availability θ_i to improve the throughput. However, when θ_i's are all similar, selecting from all channels will not result in significant throughput loss. Instead, it can reduce collision events among SUs. Thus, in this case, the policy in [93] performs slightly better than our policy.

In summary, the numerical results show that the effectiveness of our proposed policy will not be impacted by different channel availability distributions compared with the existing access policies. Designing such a learning and access policy that work well for all mean channel availability distribution of the primary channels is practically of great interest.

5.7 Summary

In this chapter, we considered the problem of decentralized online learning and channel access in a cognitive radio network through a game theoretic approach. We proved that this game is an exact potential game, therefore can at least achieve a pure Nash equilibrium point. We aimed at designing an adaptive policy that can effectively respond to different channel availability scenarios across primary channels. To do this, each SU probabilistically selects one of its estimated M-best channels to access, and

the SU updates the channel selection probability based on collision events. Our proposed adaptive policy consists of two underlying distributed learning algorithms, one is UCB-based learning from sensing history on the primary channel availability, and the other is SLA-based learning from collision history on channel selections among SUs to avoid further collision. We further proved the convergence of our proposed adaptive learning algorithm towards Nash equilibrium point of the game. Numerical results showed the effectiveness of our proposed adaptive policy in various distributions of mean availabilities across primary channels, as compared with other existing policies.

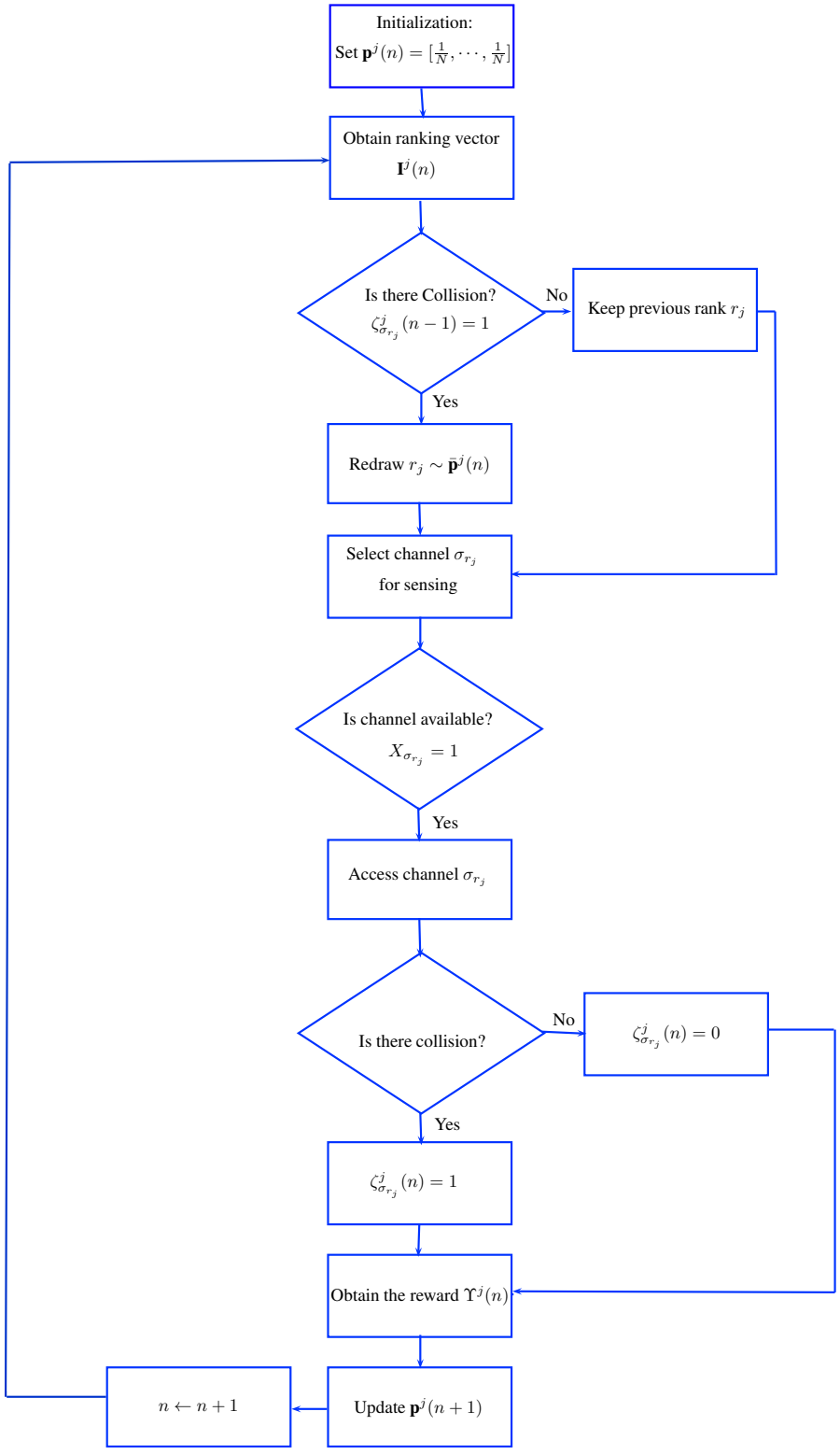

Figure 5.1: Distributed adaptive learning and access policy: For SU j at time slot n

Algorithm 6 Distributed adaptive learning and access policy: //For SU j at time slot n

1: **Input**:

n: Current time slot;
r_j: the rank of channel used at time slot $n-1$;
$\zeta_i^j(n-1) \in \{0,1\}$: the collision acknowledgment for user j on channel i.
p_{th}: threshold for checking convergence of $p_i^j(n)$

2: **if** $n == 0$ **then** //initialization
Set $\mathbf{p}^j(n) = [\frac{1}{N}, \cdots, \frac{1}{N}]$.
end if

3: *Select channel to sense and access*:

i) Obtain its ranking vector $\mathbf{I}^j(n)$.

ii) **if** $\zeta_{\sigma_{r_j}}^j(n-1) == 1$ **then** //collision
Redraw $r_j \sim \bar{\mathbf{p}}^j(n)$ as in (5.3.3)
else // no collision
Keep previous rank r_j
end if

iii) Select channel σ_{r_j} for sensing.
If $X_{\sigma_{r_j}} = 1$, **then** //channel is available
Access channel σ_{r_j}.
end if

iv) Update acknowledgement
If collision **then**
$\zeta_{\sigma_{r_j}}^j(n) = 1$
else
$\zeta_{\sigma_{r_j}}^j(n) = 0$.
end if

4: *Update Channel Selection Probability vector*:

i) Obtain the reward $\Upsilon^j(n)$ as in (5.4.2).

ii) Update $\mathbf{p}^j(n+1)$ according to the rule as in (5.3.6) and (5.3.7).

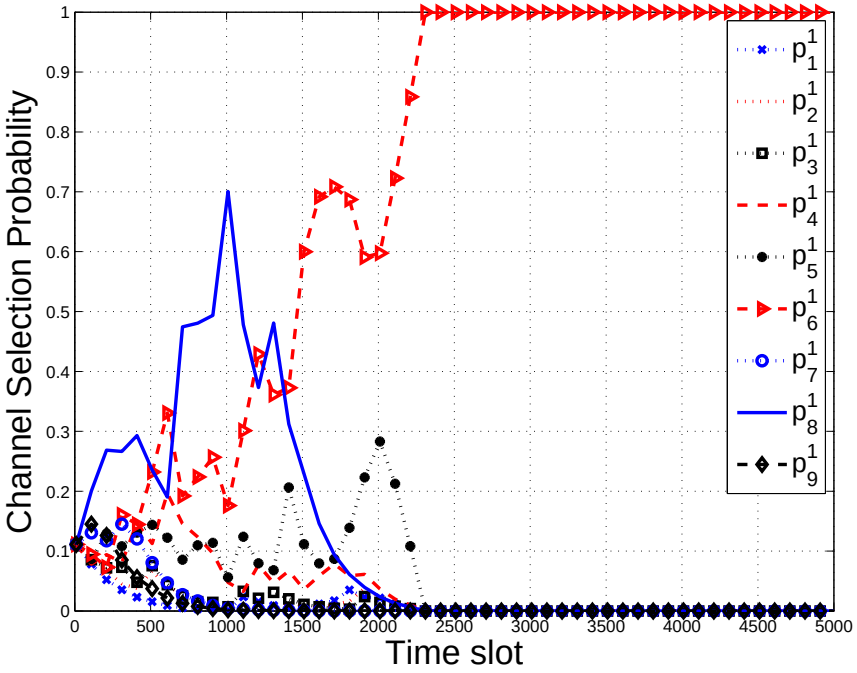

Figure 5.2: Channel selection probability vs. time slot n for SU 1 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 1, $b = 0.01$).

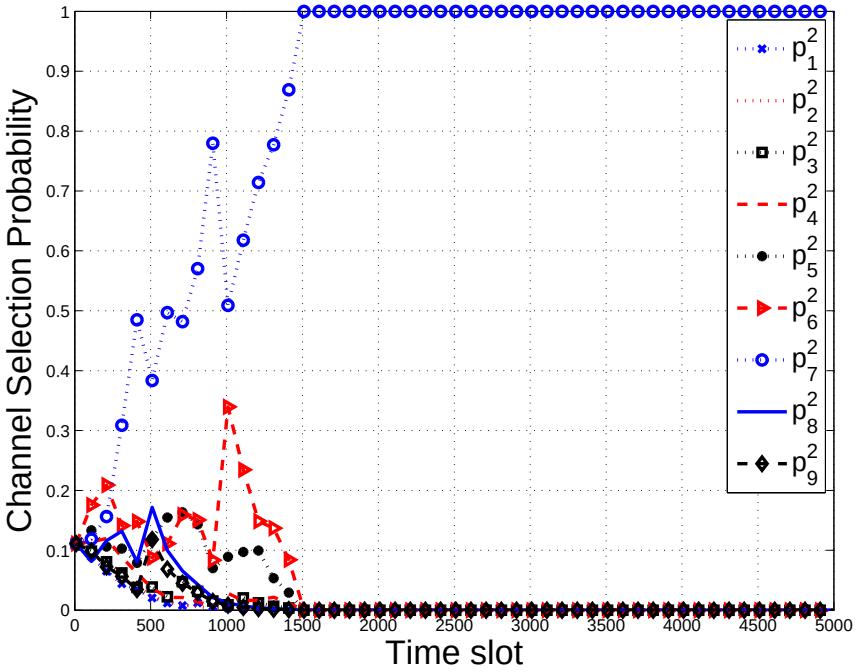

Figure 5.3: Channel selection probability vs. time slot n for SU 2 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 1, $b = 0.01$).

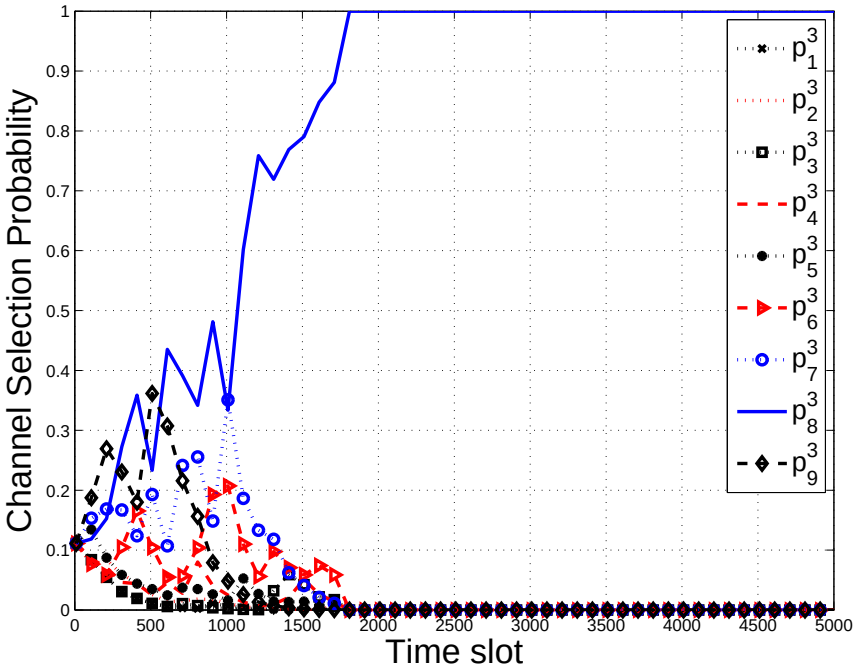

Figure 5.4: Channel selection probability vs. time slot n for SU 3 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 1, $b = 0.01$).

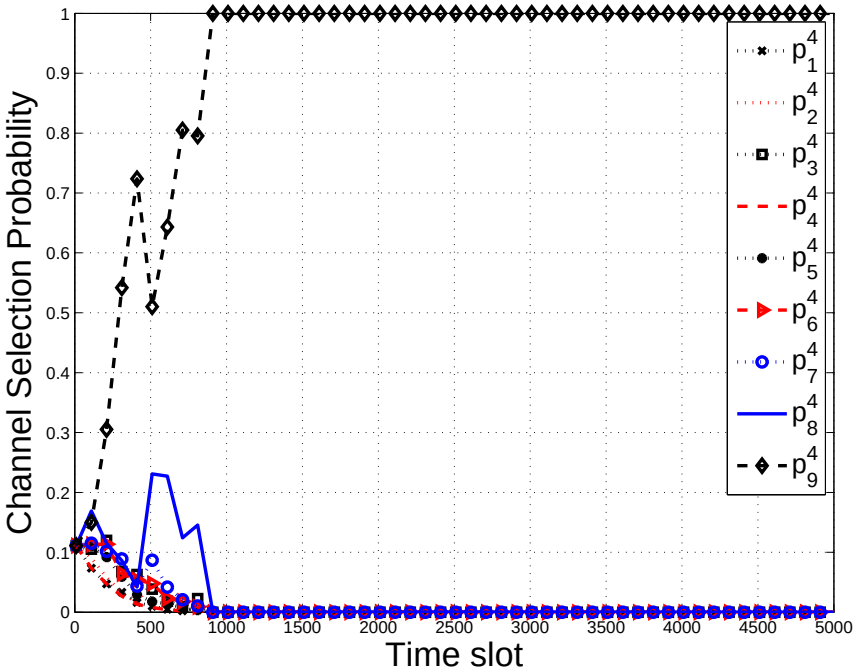

Figure 5.5: Channel selection probability vs. time slot n for SU 4 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 1, $b = 0.01$).

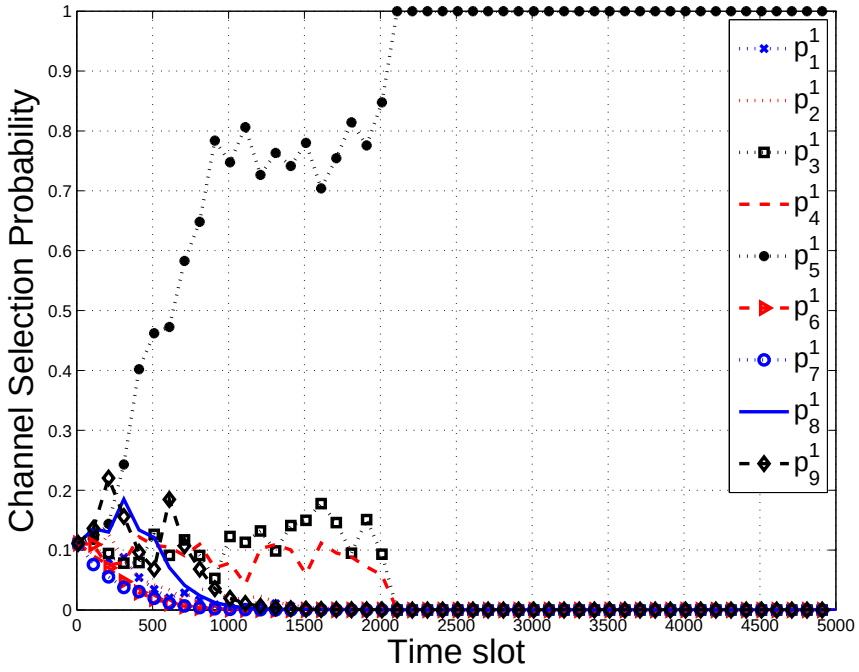

Figure 5.6: Channel selection probability vs. time slot n for SU 1 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 2, $b = 0.01$).

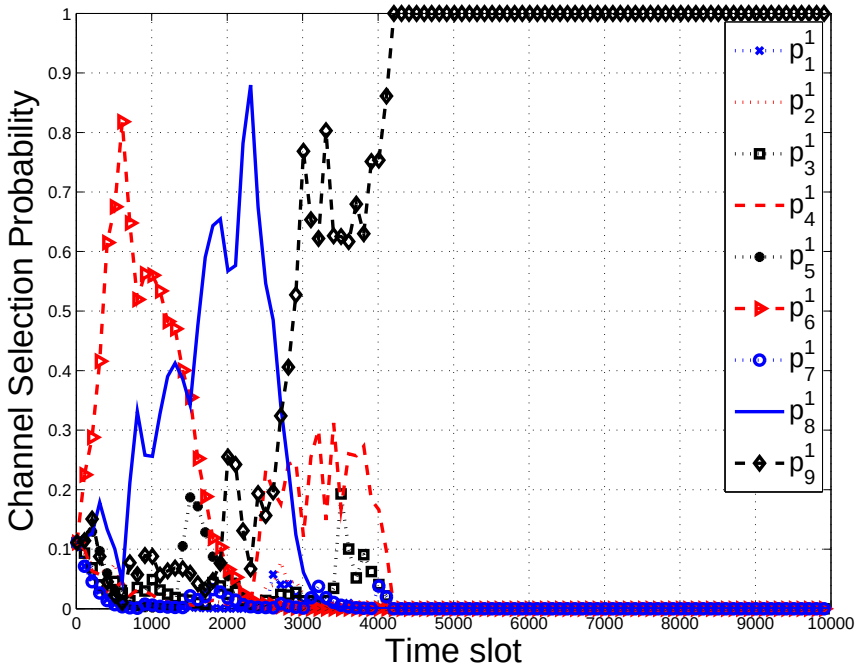

Figure 5.7: Channel selection probability vs. time slot n for SU 2 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 2, $b = 0.01$).

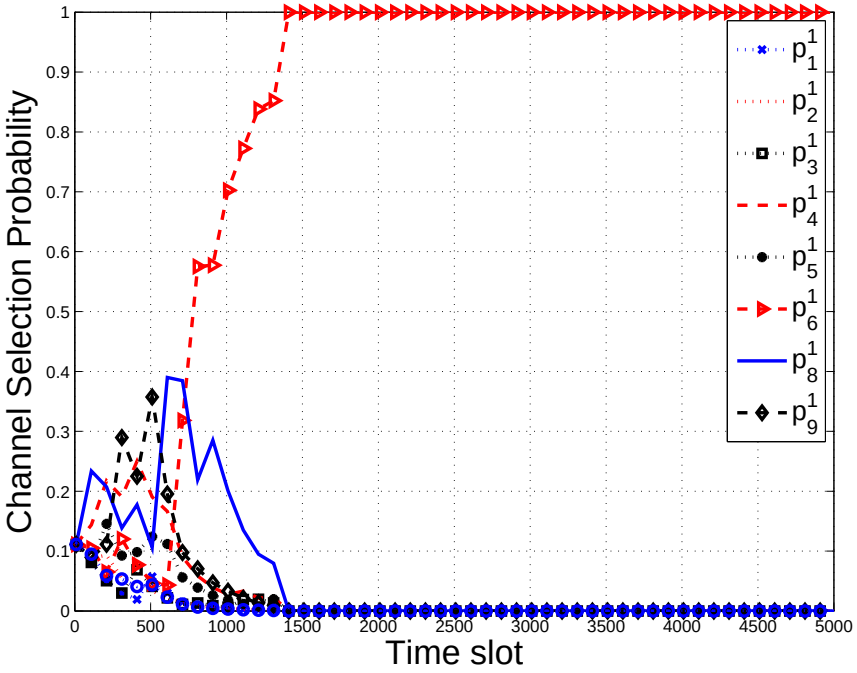

Figure 5.8: Channel selection probability vs. time slot n for SU 3 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 2, $b = 0.01$).

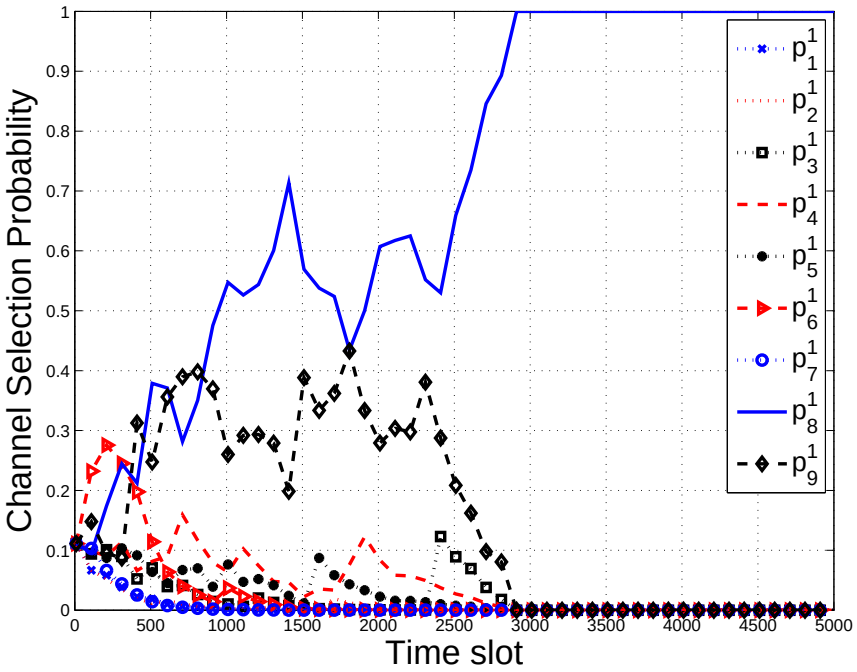

Figure 5.9: Channel selection probability vs. time slot n for SU 4 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 2, $b = 0.01$).

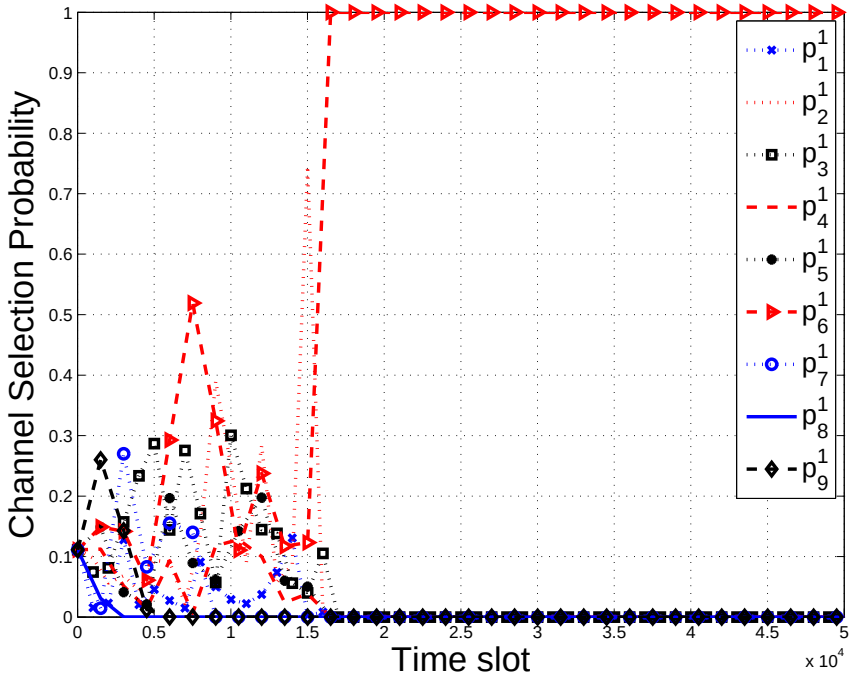

Figure 5.10: Channel selection probability vs. time slot n for SU 1 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 3, $b = 0.01$).

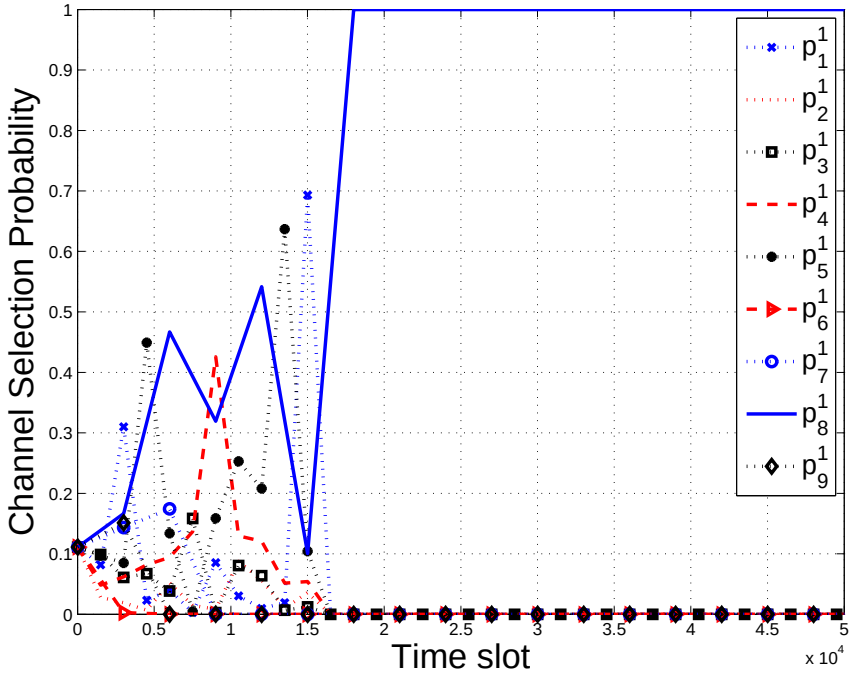

Figure 5.11: Channel selection probability vs. time slot n for SU 2 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 3, $b = 0.01$).

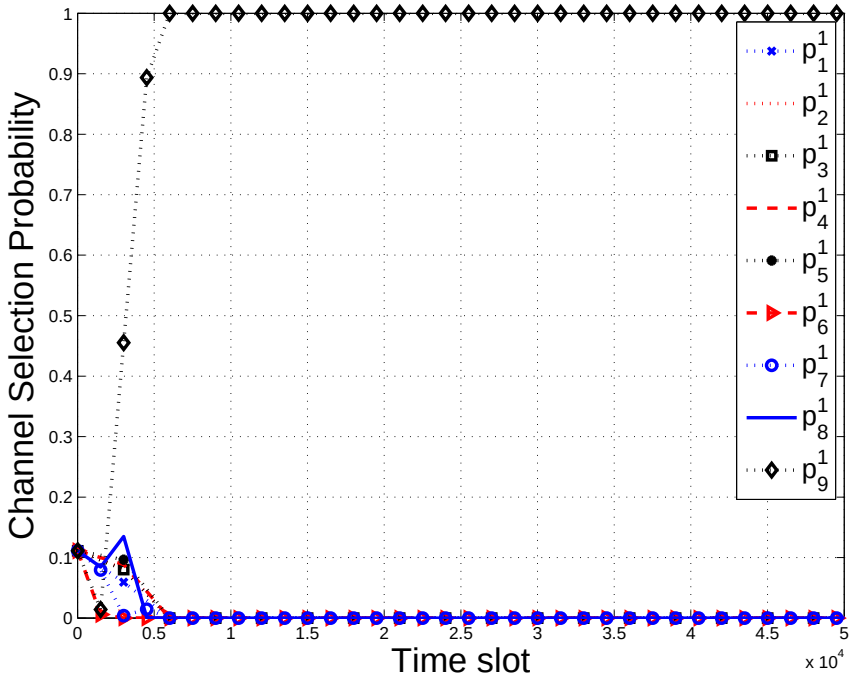

Figure 5.12: Channel selection probability vs. time slot n for SU 3 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 3, $b = 0.01$).

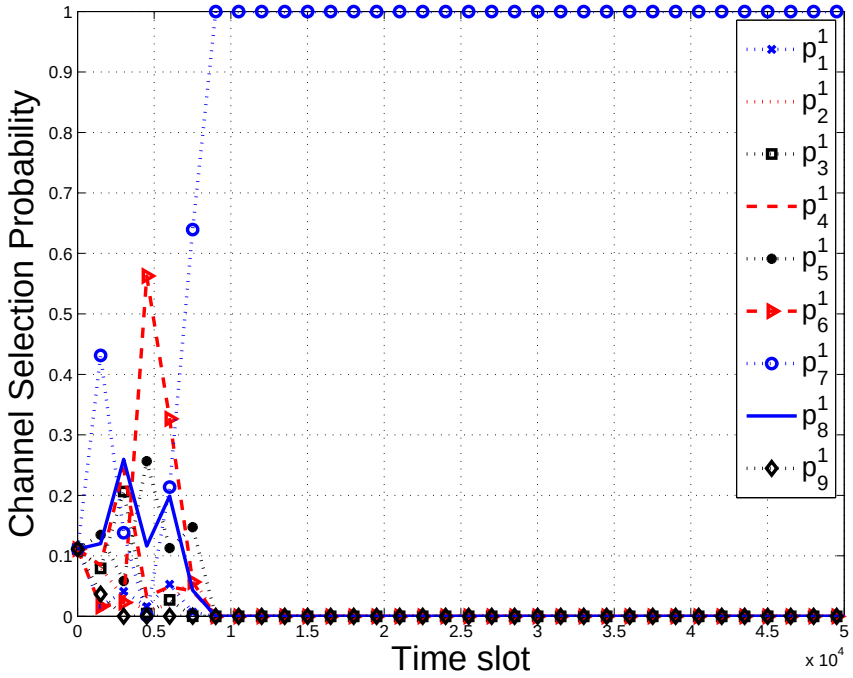

Figure 5.13: Channel selection probability vs. time slot n for SU 4 ($M = 4$, $N = 9$, $\boldsymbol{\theta}$: case 3, $b = 0.01$).

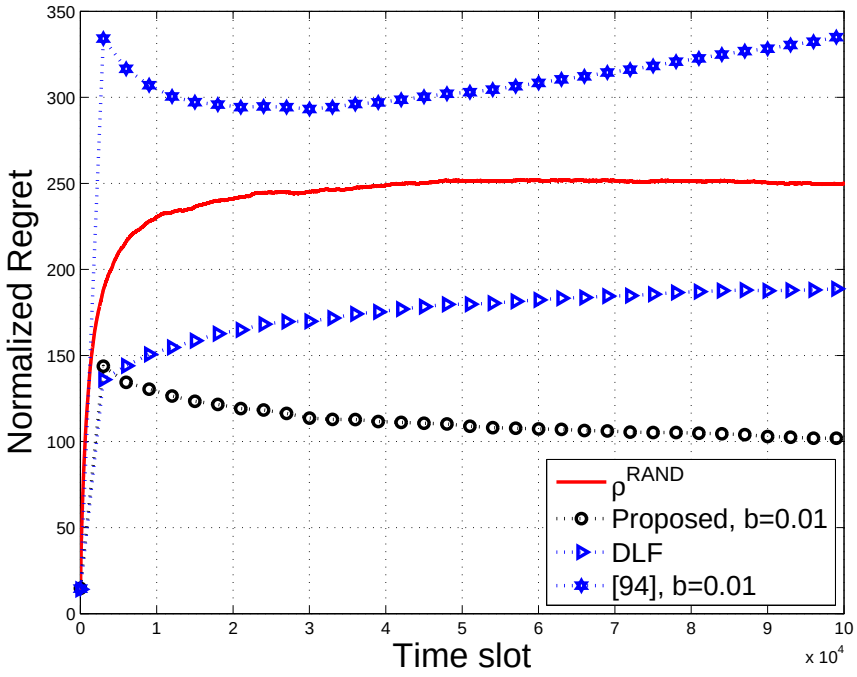

Figure 5.14: Normalized regret vs. time slot n ($\boldsymbol{\theta}$: case 1, $M = 4$, $N = 9$).

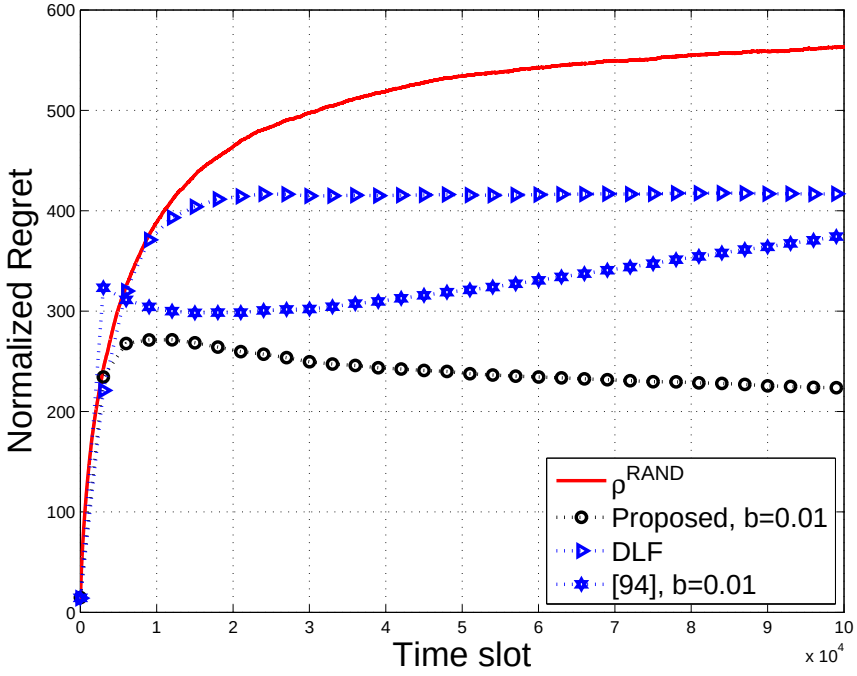

Figure 5.15: Normalized regret vs. time slot n ($\boldsymbol{\theta}$: case 2, $M = 4$, $N = 9$).

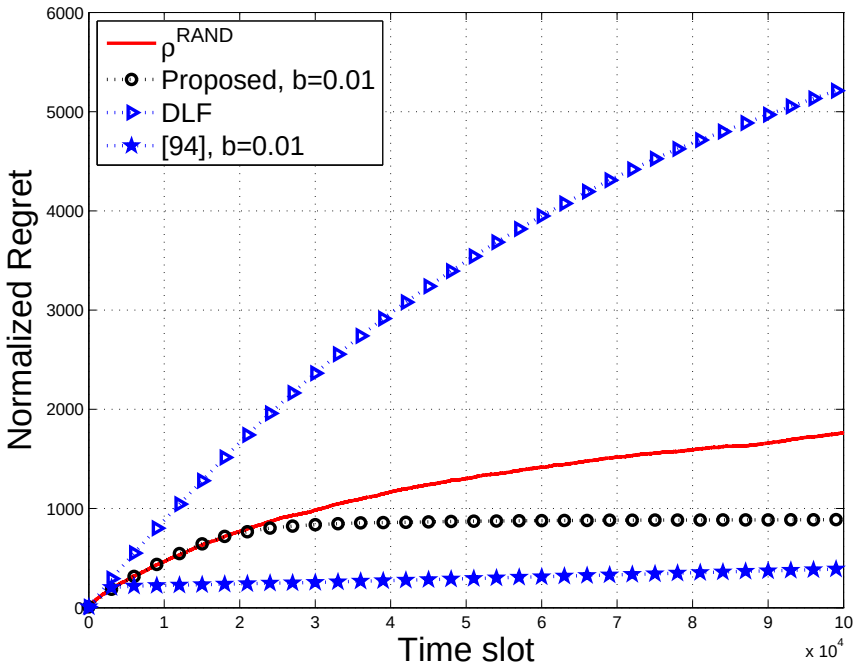

Figure 5.16: Normalized regret vs. time slot n ($\boldsymbol{\theta}$: case 3, $M = 4$, $N = 9$).

Chapter 6

Conclusion and Future Research

6.1 Conclusion

In this thesis, we first considered the auction-based approaches for dynamic spectrum access where primary channel availability statistics are unknown to the SUs. Since primary channels are with distinct availability statistics, thus bidding such channels among SUs can be viewed as bidding multiple heterogenous objects. The UD auction was first applied and then the instantaneous link condition of each SU over the primary channels for its throughput maximization was explored. To avoid accessing less available primary channels, we further proposed the LBUD auction, in which distributed learning of the primary channels at each SU is performed and incorporated in the auction mechanism. To maximize SUs' throughputs, our proposed LBUD auction explores both channel availability statistics and instantaneous link gains of the SUs. Applying the LBUD auction, communication overhead of the required bidding

data is reduced compared with the UD auction. Combination of the primary channel availabilities and the secondary link conditions is not investigated in the existing works. We showed that the proposed LBUD auction is DSIC. To improve convergence speed of the iterative procedure in the auction, we further proposed an adaptive price increment algorithm. Our simulation results showed the effectiveness of our proposed auction mechanism in terms of the throughput gain.

In addition to auction approach, we also studied multi-armed bandit framework as the second approach for designing decentralized online learning and channel access mechanismsin cognitive radio network. We developed a truly distributed dynamic spectrum access mechanism where both secondary user population and mean channel availabilities are unknown to the SUs. To do so, we design a thresholding mechanism for online estimation of secondary user population over time, we extend an existing policy [9] to the scenario with unknown user population. The proposed thresholding method dynamically adjusts the threshold for updating the secondary user population, using virtual systems built upon the current estimates of mean channel availabilities. Our proposed algorithm allows both overestimation and underestimation in estimating the secondary user poulation over time, and therefore can be applied in a dynamic environment with population change. Based on the existing distributed access policy, the ρ^{RAND} policy [9] , we further proposed an adaptive decentralized access policy which adjusts the distributed coordination mechanism among SUs by adaptively changing the "perceived population" at each SU to reduce collisions at different learning accuracy stages. We design a metric that measures the level of learning accuracy and use that as an indicator to adjust the "perceived population" by each SU. Our numerical results showed that our proposed adaptive policy improves

the scaling constant of the normalized regret and provided substantial improvement over the ρ^{RAND} policy. We studied the problem on how the availability distribution of primary channels affect the performance of distributed learning and access policies. Aiming at designing an algorithm that adapts to different channel availability distribution conditions, we first extended the recently proposed BLA algorithm to distributed online learning, and formed learning and access policies modified from existing access policies. By analyzing the distributed access collision mechanism offered by the ρ^{RAND} and DLF policies, we identify how different mean channel availability distributions affect the effectiveness of each policy. Based on this, we developed dynamic switching mechanism for online learning and access algorithm (*i.e.,* DSLA) that adapts to different channel availability distribution conditions. The switching is based on a closeness factor we proposed to determine which learning or access policies is most effective for a given primary channel condition. Simulation studies showed that our proposed DSLA policy is effective provides good performance for a wide range of θ_i's distributions.

As the third approach, we considered a game theoretic approach for designing a decentralized online learning and channel access in a cognitive radio network aiming at designing an adaptive policy that can effectively respond to different channel availability scenarios across primary channels. We proved that this game is an exact potential game, thus can at least achieve a pure Nash equilibrium point. In our design, each SU probabilistically selects one of its estimated M-best channels to access, and the SU updates the channel selection probability based on collision events. Two underlying distributed learning algorithms are considered in our adaptive policy design, one is UCB-based learning from sensing history on the primary channel

availability, and the other is SLA-based learning from collision history on channel selections among SUs to avoid further collision. We further proved the convergence of our proposed adaptive learning algorithm towards Nash equilibrium point of the game. Numerical results showed the effectiveness of our proposed adaptive policy in various distributions of mean availabilities across primary channels, as compared with other existing policies.

6.2 Future Research

We now discuss some potential directions of future research on designing online learning and channel access in a cognitive radio network considered in this thesis.

Considering auction formulation, the approach proposed in this thesis might be further developed to the scenario where the secondary user population M is unknown to the SUs. Specifically, it will be interesting to develop an auction based policy in a dynamic environment. The approach proposed in this thesis may be further developed to combinatorial auction design, auctions where bidders bid on combinations of the objects. This study can also be extended to a truly decentralized scenario. Considering SLA formulation, our proposed approach might be further developed to an adaptive step size b.

Appendices

Appendix A

Proofs in Chapter 3

A.1 Proof of Proposition A.1

We consider the UD auction procedure at time slot n, as described in Section 3.3. The convergence of the iterative procedure under unit price increment, *i.e.,* $\Delta P_{n,l} = 1$, has already been shown in [8] in the integer-valued bidding scenario. First consider the integer-valued scenario. Consider the lth and $(l+1)$th iterations under this unit price increment:

At Iteration l For SU j, following Steps 3 and 4 of the UD auction, the auctioneer obtains the demand set $\mathcal{D}^j(\mathbf{P}^l(n))$ in (3.3.3), and the payoff difference $\Delta\rho^j(n,l)$ in (3.4.10). In Step 4.b), assume that there exist overdemanded sets among the demand sets $\{\mathcal{D}^1(\mathbf{P}^l(n)), \cdots, \mathcal{D}^M(\mathbf{P}^l(n))\}$. The auctioneer updates the price $P_i^l(n)$ by (3.3.7), for $i \in \mathcal{D}^{\min}(\mathbf{P}^l(n))$, *i.e.,* those channels in the *minimal overdemanded* set.

Define $\mathcal{S}_{\min} = \{j : \mathcal{D}^j(\mathbf{P}^l(n)) = \mathcal{D}^{\min}(\mathbf{P}^l(n)), \forall j\}$ as the set of SUs whose demanded set is the minimal overdemanded set. Note that since k_1^j in (3.4.8) is the channel with the highest current payoff for SU j, we have $k_1^j \in \mathcal{D}^{\min}(\mathbf{P}^l(n))$ if $j \in \mathcal{S}_{\min}$.

Assume $\Delta\rho^j(n,l) = L_j \geq 1$. Let $j^* \in \mathcal{S}_{\min}$ be the SU with

$$L_{j^*} = \min_{j \in \mathcal{S}_{\min}} L_j. \tag{A.1.1}$$

The corresponding price and the payoffs are updated as follows

$$P^{l+1}_{k_1^{j^*}}(n) = P^l_{k_1^{j^*}}(n) + 1, \tag{A.1.2}$$

$$\rho^j_{k_1^{j^*}}(n, l+1) = \rho^j_{k_1^{j^*}}(n,l) - 1, \tag{A.1.3}$$

$$\rho^j_{k_2^{j^*}}(n, l+1) = \rho^j_{k_2^{j^*}}(n,l), \tag{A.1.4}$$

where (A.1.2) follows (3.3.7) with $\Delta P_{n,l} = 1$, and (A.1.3) is due to (3.4.7) and (A.1.2). For (A.1.4), since $L_{j^*} \geq 1$, we know $k_2^{j^*} \notin \mathcal{D}^{\min}(\mathbf{P}^l(n))$.

At Iteration $l+1$ The same procedure in Steps 3 and 4 follows. From (A.1.3) and (A.1.4), we have the payoff difference as

$$\begin{aligned}\rho^j_{k_1^{j^*}}(n, l+1) - \rho^j_{k_2^{j^*}}(n, l+1) &= \rho^j_{k_1^{j^*}}(n,l) - \rho^j_{k_2^{j^*}}(n,l) - 1 \\ &= L_{j^*} - 1. \end{aligned} \tag{A.1.5}$$

In this case, channel $k_1^{j^*}$ still remains the channel with the highest current payoff for the SU j^*. As a consequence, the minimal overdemanded set does not change, and we have

$$P^{l+2}_{k_1^{j^*}}(n) = P^{l+1}_{k_1^{j^*}}(n) + 1.$$

The above procedure will be repeated L_{j^*} times till at the iteration $l + L_{j^*}$, we have

$$\rho^j_{k_2^{j^*}}(n, l + L_{j^*}) = \rho^j_{k_1^{j^*}}(n, l + L_{j^*}). \tag{A.1.6}$$

Based on the definition of the demand set in (3.3.3), the above will lead to a change of the demand set $\mathcal{D}^{j^*}(\mathbf{P}^{l+1}(n))$ for SU j^*, as compared to that in iteration l. As a result, this may result in a change of the exclusive demanders $\mathcal{B}^{\mathrm{E}}(\mathcal{D}^{j^*}(\mathbf{P}^{l+1}(n)))$ for $\mathcal{D}^{j^*}(\mathbf{P}^{l+1}(n))$, and consequently a change of the *minimal overdemanded* set $\mathcal{D}^{\min}(\mathbf{P}^{l+1}(n))$.

Now, we apply our proposed adaptive price mechanism for SU j at time slot n. At iteration l, we assumed that $\Delta\rho^j(n,l) = L_j$, therefore, we have $\Delta P_{n,l} = L$, where

$$L = \min_{1\le j\le M} L_j.$$

Assume there are overdemanded sets among the demand sets $\{\mathcal{D}^j(\mathbf{P}^{l+1}(n))\}$. The auctioneer updates the price $P_i^l(n)$ in the price vector $\mathbf{P}^{l+1}(n)$, for $i \in \mathcal{D}^{\min}(\mathbf{P}^{l+1}(n))$. Again, for $j \in \mathcal{S}_{\min}$, we know that $k_1^j \in \mathcal{D}^{\min}(\mathbf{P}^{l+1}(n))$. Find SU $j^* \in \mathcal{S}_{\min}$, satisfying (A.1.1). We have two cases:

i) If $L_{j^*} = L$: We have

$$P_{k_1^{j^*}}^{l+1}(n) = P_{k_1^{j^*}}^{l}(n) + \Delta P_{n,l}, \tag{A.1.7}$$

$$\rho_{k_1^{j^*}}^{j}(n, l+1) = \rho_{k_2^{j^*}}^{j^*}(n, l+1). \tag{A.1.8}$$

Compare (A.1.6) and (A.1.8) with the unit price increment and adaptive price increment, respectively, we see that the latter reaches the same result of the former in just one iteration.

ii) If $L_{j^*} > L$: We have

$$\rho_{k_2^{j^*}}^{j}(n, l + \lceil\frac{L_{j^*}}{L}\rceil) \ge \rho_{k_1^{j^*}}^{j}(n, l + \lceil\frac{L_{j^*}}{L}\rceil). \tag{A.1.9}$$

Compare (A.1.6) and (A.1.9) with the unit price increment and adaptive price increment, respectively, we see that the latter reaches the same result of the former faster in $\lceil\frac{L_{j^*}}{L}\rceil$ iterations.

Since the convergence of the UD auction with the unit price increment has been shown [8], it follows that the UD auction with the adaptive price increment is also convergent.

The above analysis can be easily applied to the real-valued case. To see this, we note that any real-valued quantity can be converted to an integer value by multiplying it by an integer. Then, the iterative process follows with guaranteed convergence.

The above proof can be straightforwardly applied to the LBUD auction to show the convergence under adaptive price increment. The only difference lies in the fact that in the LBUD auction, the demand set is considered as in (3.4.4).

A.2 Proof of Proposition A.2

To prove the proposition, we first convert the LBUD auction in the format of the UD auction. Then we show that, as the underlying learning of primary channels improves over time slot n, the auction becomes DSIC.

In the LBUD auction, each SU j observes its current valuation, *i.e.,* $R_i^j(n)$ for channel $i \in \mathcal{C}_M^j(n)$, decides the corresponding bid $m_i^j(n)$. The bids for each SU are submitted for its estimated M-best channels. Thus, the set of bids contains the information on the set of M channels considered.

Let $\mathcal{C}_M$ denotes the set of M-best channels. The valuation of each channel for SU j contains two facts: a) instantaneous rate $R_i^j(n)$ on channel i , and b) whether or not $i \in \mathcal{C}_M$. Thus, bidding truthfully (*i.e.,* a bid equals to the valuation) means truthfully selecting the M-best channels and truthfully reporting the instantaneous rate over each of these channels.

Now we design another UD auction. We construct a modified bid as follows

$$\bar{m}_i^j(n) = \begin{cases} m_i^j(n), & i \in \hat{\mathcal{C}}_M^j(n) \\ -\infty, & \text{otherwise} \end{cases} \tag{A.2.1}$$

where $\hat{\mathcal{C}}_M^j(n)$ denotes any set of M channels in $\mathcal{C}$. From above, we see that selecting a different set of M channels is equivalent to setting different bid for each channel, resulting in a different set of bids. Based on (A.2.1), we have a corresponding modified valuation $\bar{R}_i^j(n)$ as

$$\bar{R}_i^j(n) = \begin{cases} R_i^j(n), & i \in \mathcal{C}_M \\ -\infty, & \text{otherwise} \end{cases}. \tag{A.2.2}$$

In the LBUD auction, each SU j has its own estimated M-best channel $\mathcal{C}_M^j(n)$, thus its bid is given by

$$\hat{R}_i^j(n) = \begin{cases} R_i^j(n), & i \in \mathcal{C}_M^j(n) \\ -\infty, & \text{otherwise} \end{cases}. \tag{A.2.3}$$

Each SU j uses this modified bid $\hat{R}_i^j(n)$ to bid for every channel $i \in \mathcal{C}$. At the auctioneer, the demand set for each SU j is denoted as $\hat{\mathcal{D}}^j(\mathbf{P}^l(n))$, which is similar to (3.3.3) and is given by

$$\hat{\mathcal{D}}^j(\mathbf{P}^l(n)) = \left\{ \arg\max_{i \in \mathcal{C}} (\hat{R}_i^j(n) - P_i^l(n)) \right\}. \tag{A.2.4}$$

The auctioneer will carry out the assignment using the UD auction mechanism described in Section 3.3. Using (A.2.2), we examine (3.4.4) and (A.2.6) and can see that

$$\hat{\mathcal{D}}^j(\mathbf{P}^l(n)) = \mathcal{D}_M^j(\mathbf{P}^l(n)). \tag{A.2.5}$$

As mentioned in the main text, the distributed learning of M-best channels by (3.4.2) and (5.3.1) under the UCB1 algorithm has been shown to be order-optimal in terms of the learning rate over time [44]. Specifically, as the time slot $n \to \infty$, the probability of not selecting the true M-best channel is

$$\text{Prob}(C_M^j(n) \neq C_M) = \mathcal{O}\left(\frac{\log n}{n}\right) \to 0.$$

Thus, we have $\hat{R}_i^j(n) \to \bar{R}_i^j(n)$ in probability. In other words, the bids for each SU j converges to the valuations of M-best channels in probability. Consequently, let

$$\bar{\mathcal{D}}^j(\mathbf{P}^l(n)) = \left\{ \arg\max_{i \in \mathcal{C}} (\bar{R}_i^j(n) - P_i^l(n)) \right\}. \tag{A.2.6}$$

From (A.2.5), we have $\mathcal{D}_M^j(\mathbf{P}^l(n)) \to \bar{\mathcal{D}}^j(\mathbf{P}^l(n))$ in probability.

Thus, in the long run as $n \to \infty$, the LBUD auction is essentially equivalent to the new UD auction we constructed. Since the UD auction is dominant strategy incentive compatible, it follows that the proposed LBUD also carries such a property.

Appendix B

Proofs in Chapter 5

B.1 Proof of Proposition B.1

We define a potential function known as Rosenthal's potential function [100] for our game $\mathcal{G}_p$ as follows:

$$\mathcal{P}(a_j, a_{-j}; n) \triangleq \sum_{i=1}^{N} \sum_{k=1}^{m_i(n)} \psi_i(k), \tag{B.1.1}$$

where $\psi_i(k)$ is defined in (5.4.5).

Let assume $a_j(n)$ and $\tilde{a}_j(n)$ as two different actions which SU j may take from the set of possible actions, *i.e.* $a_j(n), \tilde{a}_j(n) \in \mathcal{A}_j(n)$ and $a_j(n) \neq \tilde{a}_j(n)$. We define $\hat{\mathcal{C}}_M^j(n)$ as

$$\hat{\mathcal{C}}_M^j(n) \triangleq \mathcal{C}_M^j(n) \backslash \{a_j(n), \tilde{a}_j(n)\}. \tag{B.1.2}$$

Then, the potential function in (B.1.1) can be rewritten by

$$\begin{aligned}
&\mathcal{P}(a_j, a_{-j}; n) \\
&= \sum_{i=1}^{N} \sum_{k=1}^{m_i(n)} \psi_i(k) \\
&= \sum_{i \in \hat{\mathcal{C}}_M^j(n)} \sum_{k=1}^{m_i(n)} \psi_i(k) + \sum_{k=1}^{m_{a_j}(n)} \psi_{a_j}(k) + \sum_{k=1}^{m_{\tilde{a}_j}(n)} \psi_{\tilde{a}_j}(k).
\end{aligned}$$

For SU j changes its channel access to $\tilde{a}_j(n) \in \mathcal{A}_j(n)$, the potential function is given by

$$\begin{aligned}
&\mathcal{P}(\tilde{a}_j, a_{-j}; n) \\
&= \sum_{i=1}^{N} \sum_{k=1}^{m_i(n)} \psi_i(k) \\
&= \sum_{i \in \hat{\mathcal{C}}_M^j(n)} \sum_{k=1}^{m_i(n)} \psi_i(k) + \sum_{k=1}^{m_{a_j(n)}-1} \psi_{a_j}(k) + \sum_{k=1}^{m_{\tilde{a}_j(n)}+1} \psi_{\tilde{a}_j}(k).
\end{aligned}$$

Thus, when SU j changes its channel access from $a_j(n)$ to $\tilde{a}_j(n)$, the deviation in the potential function is given by

$$\begin{aligned}
&\mathcal{P}(\tilde{a}_j, a_{-j}; n) - \mathcal{P}(a_j, a_{-j}; n) \\
&\qquad = \psi_{\tilde{a}_j}(m_{\tilde{a}_j}(n) + 1) - \psi_{a_j}(m_{a_j}(n)).
\end{aligned} \tag{B.1.3}$$

For SU j, the change in its payoff by switching from $a_j(n)$ to $\tilde{a}_j(n)$ can be obtained by

$$\begin{aligned}
&u_j(\tilde{a}_j, a_{-j}; n) - u_j(a_j, a_{-j}; n) \\
&\qquad = \psi_{\tilde{a}_j(n)}(m_{\tilde{a}_j}(n) + 1) - \psi_{a_j(n)}(m_{a_j}(n)).
\end{aligned} \tag{B.1.4}$$

By (B.1.3) and (B.1.4), it follows that the property in (5.4.6) holds. Therefore, the game $\mathcal{G}_p$ is an exact potential game.

B.2 Proof of Proposition B.2

Assume that SU j chooses a pure strategy of selecting channel i, and any other SU $s \in \mathcal{S}$, $s \neq j$, takes a mixed strategy $\mathbf{p}^s(n)$. Let $\mathbf{P}_{-j}(n) \triangleq \{\mathbf{p}^s(n) : s \in \mathcal{S}\backslash\{j\}\}$ be the set of channel probability selection vectors of the SU j's opponents at time slot n, and let $\mathbf{e}_i$ be a unit vector with the i^{th} entry being 1 and the rest 0's. Then, the expected throughput of SU j at time slot n, denoted by $\bar{u}_j(\mathbf{e}_i, \mathbf{P}_{-j}; n)$, is given by

$$\bar{u}_j(\mathbf{e}_i, \mathbf{P}_{-j}; n) = \sum_{\substack{a_s(n)\in\mathcal{A}_s(n) \\ \forall\ s\neq j}} u_j(i, \{a_1, ..., a_{j-1}, a_{j+1}, ..., a_M\}; n) \prod_{\substack{s=1 \\ s\neq j}}^{M} p^s_{a_s}(n). \tag{B.2.1}$$

a_j and a_{-j} are random variables. We define $\tilde{\mathcal{P}}(\mathbf{p}^j, \mathbf{P}_{-j}; n)$ is a function of random variables a_j and a_{-j} which are selected based on the channel selection probability distributions $\mathbf{p}^j$ and $\mathbf{P}_{-j}$ respectively. Note that the potential function defined in (B.1.1) is the realization of the $\tilde{\mathcal{P}}(\mathbf{p}^j, \mathbf{P}_{-j}; n)$. We define $X(\mathbf{p}^j, \mathbf{P}_{-j}; n) : \mathbf{P} \to R$ as the expected function $\tilde{\mathcal{P}}(\mathbf{p}^j, \mathbf{P}_{-j}; n)$ by SU j

$$X(\mathbf{p}^j, \mathbf{P}_{-j}; n) \triangleq E[\tilde{\mathcal{P}}(\mathbf{p}^j, \mathbf{P}_{-j}; n)]. \tag{B.2.2}$$

Then, for SU j taking a pure strategy $\mathbf{p}^j(n) = \mathbf{e}_i$, we have

$$X(\mathbf{e}_i, \mathbf{P}_{-j}; n) = \sum_{\substack{a_s(n)\in\mathcal{A}_s(n) \\ \forall\ s\neq j}} \mathcal{P}(i, \{a_1, ..., a_{j-1}, a_{j+1}, ..., a_M\}; n) \prod_{\substack{s=1 \\ s\neq j}}^{M} p^s_{a_s}(n), \tag{B.2.3}$$

where $\mathcal{P}(i, a_{-j}; n)$ is the defined potential function as in (B.1.1).

From (5.4.6) and (B.2.1), for SU j changes its selection from channel i to i', we have

$$X(\mathbf{e}_i, \mathbf{P}_{-j}; n) - X(\mathbf{e}_{i'}, \mathbf{P}_{-j}; n) = \bar{u}_j(\mathbf{e}_i, \mathbf{P}_{-j}; n) - \bar{u}_j(\mathbf{e}_{i'}, \mathbf{P}_{-j}; n). \tag{B.2.4}$$

We then prove our result using the result in [93] which is stated below.

Theorem 1. [93, Theorem 5] Suppose that there is a non-negative function $X(\mathbf{p}^j, \mathbf{P}_{-j}; n)$: $\mathbf{P} \to R$ for some positive constant $c > 0$ such that

$$X(\mathbf{e}_i, \mathbf{P}_{-j}; n) - X(\mathbf{e}_{i'}, \mathbf{P}_{-j}; n) = c[\bar{u}_j(\mathbf{e}_i, \mathbf{P}_{-j}; n) - \bar{u}_j(\mathbf{e}_{i'}, \mathbf{P}_{-j}; n)],\ \forall j, i, i', \mathbf{P}. \tag{B.2.5}$$

Then, the SLA-based algorithm converges to a pure strategy NE of a game.

Based on the result in Theorem 1, from (B.2.4), it follows that our proposed algorithm converges to pure strategy NE points of the game $\mathcal{G}_p$.

Bibliography

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Commun. Mag.*, vol. 6, pp. 13–18, Aug. 1999.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 201–220, Feb. 2005.

[3] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 50, pp. 2127–2159, Sept. 2006.

[4] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive mac for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 589–600, Apr. 2007.

[5] Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Mag.*, vol. 24, pp. 79–89, May 2007.

[6] A. Garhwal and P. P. Bhattacharya, "A survey on dynamic spectrum access techniques for cognitive radio," *International Journal of Next-Generation Networks (IJNGN)*, vol. 3, pp. 15–32, Dec. 2011.

[7] P. P. Bhattacharya, R. Khandelwal, R. Gera, and A. Agarwal, "Smart radio spectrum management for cognitive radio," *International Journal of Distributed and Parallel Systems (IJDPS)*, vol. 2, pp. 12–24, July 2011.

[8] G. Demange, D. Gale, and M. Sotomayor, "Multi-item auctions," *Journal of Political Economy*, vol. 94, pp. 863–872, Aug. 1986.

[9] A. Anandkumar, N. Michael, K. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 731–745, Apr. 2011.

[10] Y. Gai and B. Krishnamachari, "Decentralized online learning algorithms for opportunistic spectrum access," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2011.

[11] P. Klemperer, *Auctions: Theory and Practice*. Princeton University Press, 2004.

[12] W. Vickery, "Counterspeculation, auctions, and competitive sealed tenders," *Journal of Finance*, vol. 16, pp. 8–37, Mar. 1961.

[13] Z. Han, R. Zheng, and H. V. Poor, "Repeated auctions with Bayesian nonparametric learning for spectrum access in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 10, pp. 890–900, Mar. 2011.

[14] X. Zhou and H. Zheng, "Trust: A general framework for truthful double spectrum auctions," in *Proc. IEEE Conf. on Computer Communications (INFOCOM)*, Apr. 2009.

[15] D. Niyato, E. Hossain, and Z. Han, "Dynamic spectrum access in IEEE 802.22- based cognitive wireless networks: a game theoretic model for competitive spectrum bidding and pricing," *IEEE Wireless Commun. Mag.*, vol. 16, pp. 16–23, Apr. 2009.

[16] M. N. Tehrani and M. Uysal, "Auction based spectrum trading for cognitive radio networks," *IEEE Commun. Lett.*, vol. 17, pp. 1168–1171, June 2013.

[17] L. Gao, Y. Xu, and X. Wang, "Map: Multiauctioneer progressive auction for dynamic spectrum access," *IEEE Trans. on Mobile Computing.*, vol. 10, pp. 1144–1161, Aug. 2011.

[18] Q. Shi, C. Comaniciu, and K. Jaffres-Runser, "An auction-based mechanism for cooperative sensing in cognitive networks," *IEEE Commun. Lett.*, vol. 12, pp. 3649–3661, Aug. 2013.

[19] Y. Zhu, B. Li, and Z. Li, "Designing two-dimentional spectrum auctions for mobile secondary users," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 604–613, Mar. 2013.

[20] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple player," *IEEE Trans. Signal Processing*, vol. 58, pp. 5665–5681, Nov. 2010.

[21] M. Zandi and M. Dong, "Distributed opportunistic spectrum access with unknown population," in *Proc. IEEE International Conference on Communications in China (ICCC)*, Aug. 2012.

[22] M. Zandi and M. Dong, "Learning-stage based decentralized adaptive access policy for dynamic spectrum access," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013.

[23] M. Zandi, M. Dong, and A. Grami, "Decentralized spectrum learning and access adapting to primary channel availability distribution," in *Proc. IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2013.

[24] X. Li, P. Yang, Y. Yan, L. You, S. Tang, and Q. Huang, "Almost optimal accessing of nonstochastic channels in cognitive radio networks," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Mar. 2012.

[25] V. Krishnamurthy, "Decentralized spectrum access amongst cognitive radios–an interacting multivariate global game-theoretic approach," *IEEE Trans. Signal Processing*, vol. 57, pp. 3999–4013, Oct. 2009.

[26] L. M. Law, J. Huang, and M. Liu, "Price of anarchy for congestion games in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 11, pp. 3778–3787, Oct. 2012.

[27] B. Zhang, Y. Chen, C. Wang, and K. Liu, "Learning and decision making with negative externality for opportunistic spectrum access," in *gcom = "Proc.IEEE Global Telecommn. Conf. (GLOBECOM)"*, Dec. 2012.

[28] O. Habachi, R. El-Azouzi, and Y. Hayel, "A Stackelberg model for opportunistic sensing in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 12, pp. 2148–2159, May 2013.

[29] O. Naparstek and A. Leshem, "Fully distributed optimal channel assignment for open spectrum access," *IEEE Trans. Signal Processing*, vol. 62, pp. 283–294, Jan. 2014.

[30] D. Bertsekas, "A distributed algorithm for the assignment problem," *in Lab. for Inform, and Decision Syst. Working Paper. Cambridge, MA, USA: Massachusetts Inst. Technol.*, 1979.

[31] V. Krishna, *Auction Theory*. Elsevier Science, 2002.

[32] E. H. Clarke, "Multipart pricing of public goods," *Public Choice*, vol. 11, pp. 17–33, 1971.

[33] T. Groves, "Incentives in teams," *Economietrica: Journal of Econometric Society*, vol. 41, pp. 617–631, July 1973.

[34] S. Dobzinski, R. Lavi, and N. Nisan, "Multi-unit auctions with budget limits," *Journal of Game and Economic Behavior*, vol. 74, pp. 486–503, Mar. 2012.

[35] I. Ashlagi, M. Braverman, and A. Hassidim, "Ascending unit demand auctions with budget limits," *Working Paper*, 2009.

[36] V. Conitzer and T. Sandholm, "Failures of the VCG mechanism in combinatorial auctions and exchanges," in *Proc. 5th International Joint Conference on Autonmous Agents and Multiagents Systems (aamas)*, May 2006.

[37] L. M. Ausubel and P. Milgrom, "The lovely but lonely vickery auction," *SIEPR Discussion Paper by Stanford Institute for Economic Policy research*, Aug. 2004.

[38] G. V. der Laan and Z. Yang, "An ascending multi-item auction with financially constrained bidders," *Discussion Papers in Economics*, 2011.

[39] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Jornal of Advances in Applied Mathematics*, vol. 6, pp. 4–22, Mar. 1985.

[40] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays- part i: I.i.d. rewards," *IEEE Trans. Autom. Control*, vol. 32, pp. 968–976, Nov. 1987.

[41] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays- part ii: Markovian rewards," *IEEE Trans. Autom. Control*, vol. 32, pp. 977–982, Nov. 1987.

[42] O. C. Granmo, "A Bayesian Learning Automaton for Solving Two-Armed Bernoulli Bandit Problems," in *Proc. Seventh International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2008.

[43] S. Berg, *Solving Dynamic Bandit Problems and Decentralized Games using the Kalman Bayesian Learning Automaton*. Master's thesis, University of Adger, 2010.

[44] P. Auer, N. Cesa-Bianchi, and P. Fisher, "Finite-time analysis of the multiarmed bandit problem," *Journal of Machine Learning*, vol. 47, pp. 235–256, 2002.

[45] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Journal of Biometrika*, vol. 25, pp. 275–294, Dec. 1933.

[46] J. C. Gittins and D. M. Jones, "A dynamic allocation index for the sequential design of experiments," *Progress in Statistics*, pp. 241–266, 1974.

[47] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society*, vol. 41, pp. 148–177, 1979.

[48] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of Applied Probability*, vol. 25, pp. 287–298, Jan. 1988.

[49] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Trans. Inform. Theory*, vol. 56, pp. 5547–5567, Nov. 2010.

[50] R. R. Weber and G. Weiss, "On an index policy for restless bandits," *Journal of Applied Probability*, vol. 27, pp. 637–648, Sept. 1990.

[51] R. R. Weber and G. Weiss, "Addendum to 'on an index policy for restless bandits'," *Journal of Advances in Applied Probability*, vol. 23, pp. 429–430, June 1991.

[52] K. D. Glazebrook, D. Ruiz-Hernandez, , and C. Kirkbride, "Some indexable families of restless bandit problems," *Journal of Advances in Applied Probability*, vol. 38, pp. 643–672, Sept. 2006.

[53] S. Ahmad and M. Liu, "Multi-channel opportunistic access: a case of restless bandits with multiple plays," in *Proc. Allerton Conference on Communications, Control, and Computing.*, Oct. 2009.

[54] K. Liu, Q. Zhao, and B. Krishnamachari, "Dynamic multichannel access with imperfect dynamic multichannel access with imperfect channel state detection," *IEEE Trans. Signal Processing*, vol. 58, pp. 2795–2808, May 2010.

[55] K. Liu, Q. Zhao, and B. Krishnamachari, "Distributed learning under imperfect sensing in cognitive radio networks," in *Proc. Asilomar Conf. on Signals, Systems and computers*, Nov. 2010.

[56] J. L. Ny, M. Dahleh, and E. Feron, "Multi-uav dynamic routing with partial observations using restless bandit allocation indices," in *Proc. of the 2008 American Control Conference.*, June 2008.

[57] T. Braadland and T. Northeim, *Empirical evaluation of the Bayesian learning automaton family.* Master's thesis, University of Adger, 2009.

[58] K. Liu and Q. Zhao, "Distributed learning in cognitive radio networks: Multi-armed bandit with distributed multiple players," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Mar. 2010.

[59] A. Anandkumar, N. Michael, and A. Tang, "Opportunistic spectrum access with multiple users: Learning under competition," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Mar. 2010.

[60] Y. Gai, B. Krishnamachar, R. jain, and M. Hsieh, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *Proc. IEEE Dynamic Spectrum Access Networks (DySPAN)*, Apr. 2010.

[61] O. C. Granmo and S. Glimsdal, "A two-armed bandit based scheme for accelerated decentralized learning," in *Proc. 24th international conference on Industrial engineering and other applications of applied intelligent systems conference on Modern approaches in applied intelligence (IEA/AIE)*, June 2011.

[62] R. Agrawal, "Sample mean based index policies with o (logn) regret for the multi-armed bandit problem," *Journal of Advances in Applied probability*, vol. 27, pp. 1054–1078, Dec. 1995.

[63] R. R. Bush and F. Mosteller, *Stochastic models for learning.* New York: Wiley, 1958.

[64] K. S. Narendra and M. A. L. Thathachar, "Learning automata-a survey," *IEEE Trans. on Systems, Man, and Cybernetics.*, vol. SMC-4, July 1974.

[65] I. J. Shapiro and K. S. Narendra, "Use of stochastic automata for parameter self-optimization with multimodal performance criteria," *IEEE Trans. System Science and Cybernetics*, vol. SSC-5, pp. 352–360, Oct. 1969.

[66] I. J. Shapiro, *The Use of Stochastic automata in Adaptive Control*. Ph.D. thesis, Yale University, 1969.

[67] S. Lakshmivarahan and M. A. L. Thathachar, "Optimal nonlinear reinforcement schemes for stochastic automata," *Journal of Information Sciences*, vol. 4, pp. 121–128, Apr. 1972.

[68] S. Lakshmivarahan and M. A. L. Thathachar, "Absolutely expedient learning algorithms for stochastic automata," *IEEE Trans. on Systems, Man, and Cybernetics.*, vol. SMC-3, pp. 281–286, May 1973.

[69] B. Chandrasekaran and D. C. Shen, "Stochastic automata games," *IEEE Trans. System Science and Cybernetics*, vol. SSC-5, pp. 145–149, Apr. 1969.

[70] M. L. Tsetlin, "On the behavior of finite automata in random media," *Journal of Automation and Remote Control*, vol. 22, pp. 1210–1219, 1961.

[71] M. L. Tsetlin, "Automaton theory and modeling of biological systems," *Academic Press, New York, NY, USA*, 1973.

[72] K. S. Fu and G. J. McMurtry, "A study of stochastic automata as a model for learning and adaptive controllers," *IEEE Trans. Automat. Contr*, vol. AC11, pp. 379–387, 1966.

[73] B. Chandrasekaran and D. C. Shen, "On expediency and convergence in variable-structure automata," *IEEE Trans. System Science and Cybernetics*, vol. SSC-4, pp. 52–60, Mar. 1968.

[74] K. S. Fu, "Learning control systemsreview and outlook," *IEEE Trans. Automat. Contr*, vol. AC15, pp. 210–221, 1970.

[75] V. I. Varshavskii and I. P. Vorontsova, "On the behavior of stochastic automata with a variable structure," *Journal of Automation and Remote Control*, vol. 24, pp. 327–333, 1963.

[76] R. Viswanathan, *Learning automaton: Models and applications*. Ph.D. dissertation, Yale Univ., New Haven, CT, 1972.

[77] Y. Z. Tsypkin and A. S. Poznyak, "Finite learning automata," *Engineering cybernetics*, vol. 10, pp. 478–490, 1972.

[78] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, "Learning automata: Theory, paradigms, and applications," *IEEE Trans. on Systems, Man, and Cybernetics. Part B*, vol. 32, pp. 706–709, Dec. 2002.

[79] B. J. Oommen, "Recent advances in learning automata systems," in *Proc. 2nd International Conference on Computer Engineering and Technology (ICCET)*, Apr. 2010.

[80] M. A. L. Thathachar and P. S. Sastry, "Varieties of learning automata: An overview," *IEEE Trans. on Systems, Man, and Cybernetics. Part B*, vol. 32, pp. 711–722, Dec. 2002.

[81] K. S. Narendra and S. Lakshmivarahan, "Learning automata: A critique," *Journal of Cybernetics and Information Science*, vol. 1, pp. 53–71, 1977.

[82] S. Lakshmivarahan, *Learning Algorithms: Theory and Applications*. New York: Springer-Verlag, 1981.

[83] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Applications*. New York: Pergamon, 1994.

[84] K. Najim and A. S. Poznyak, "Multimodal searching technique based on learning automata with continuous input and changing number of actions," *IEEE Trans. on Systems, Man, and Cybernetics. Part B*, vol. 26, pp. 666–673, Aug. 1996.

[85] B. J. Oommen and E. V. D. S. Croix, "String taxonomy using learning automata," *IEEE Trans. on Systems, Man, and Cybernetics. Part B*, vol. 27, pp. 354–365, Apr. 1997.

[86] C. Unsal, P. Kachroo, and J. S. Bay, "Multiple stochastic learning automata for vehicle path control in an automated highway system," *IEEE Trans. on Systems, Man, and Cybernetics. Part A*, vol. 29, pp. 120–128, Jan. 1999.

[87] Y. Xing and R. Chandramouli, "Stochastic learning solution for distributed discrete power control game in wireless data networks," *IEEE/ACM Trans. on Networking.*, vol. 16, pp. 932–944, Aug. 2008.

[88] P. S. Sastry, V. V. Phansalkar, and M. A. L. Thathachar, "Decentralized learning of Nash Equilibria in multi-person stochastic games with incomplete information," *IEEE Trans. on Systems, Man, and Cybernetics.*, vol. 24, pp. 769–777, May 1994.

[89] T. Joshi, D. Ahuja, D. Singh, and D. O. Agrawal, "Sara: Stochastic automata rate adaptation for ieee 802.11 networks," *IEEE Trans. on Parallel and Distributed Systems.*, vol. 19, pp. 1579–1590, Nov. 2008.

[90] W. Zhong, Y. Xu, M. Tao, and Y. Cai, "Game theoretic multimode precoding strategy selection for mimo multiple access channels," *IEEE Signal Pocessing Letters*, vol. 17, pp. 563–566, June 2010.

[91] B. J. Oommen and T. D. Roberts, "Continuous learning automata solutions to the capacity assignmnet problem," *IEEE Trans. on Computers.*, vol. 49, pp. 608–620, June 2000.

[92] Y. Xu, Q. Wu, and J. Wang, "Game theoretic channel selection for opportunistic spectrum access with unknown prior information," in *Proc. International Conference on Communications (ICC)*, June 2011.

[93] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y. Yao, "Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution," *IEEE Trans. Wireless Commun.*, vol. 11, pp. 1380–1391, Apr. 2012.

[94] O. Tilak, R. Martin, and S. Mukhopadhyay, "Decentralized indirect methods for learning automata games," *IEEE Trans. on Systems, Man, and Cybernetics. Part B*, vol. 41, pp. 1213–1223, Oct. 2011.

[95] Y. Narahari, *Game Theory and Mechanism Design*. World Scientific Publishing, 2014.

[96] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, Massachussets Institute of Technology, Cambridge, Massachussets, USA, 1994.

[97] R. Gibbons, *A Primary in Game Theory*. prentice Hall, 1992.

[98] R. W. Rosenthal, "A class of games possessing pure strategy Nash equilibria," *Journal of Game Theory*, vol. 2, pp. 65–67, 1973.

[99] D. Monderer and L. S. Shapley, "Potential games," *Game and Economic Behavior*, vol. 14, pp. 124–143, 1996.

[100] B. Vcking and R. Aachen, "Congestion games: Optimization in competition," in *Proc. Algorithms and Complexity in Durham Workshop*, Sept. 2006.